Vector-Valued Markov Games

DISSERTATION

zur Erlangung des akademischen Grades

Doctor rerum naturalium
(Dr. rer. nat.)

vorgelegt

der Fakultät Mathematik und Naturwissenschaften
der Technischen Universität Dresden

von

Dipl.-Math. Piškurić Mojca

geboren am 9. September 1973 in Ljubljana

Gutachter: Prof. Dr. Vladimir V. Mazalov
Prof. Dr. Mihael Perman
Prof. Dr. rer. nat. habil. Volker Nollau
Prof. Dr. Krzystof Szajowski

Eingereicht am: 20. Juli 2000

Tag der Verteidigung: 23. April 2001

Ich möchte mich bei allen Mitgliedern des Instituts für Mathematische Stochastik der Technischen Universität Dresden bedanken, die mich in den letzten Jahren in vielfältiger Form unterstützt haben. Weiter bedanke ich mich bei allen, die mir durch vielfältige, und nicht nur Mathematik gewidmete, Gespräche (wahrscheinlich) unbewusst den richtigen Weg skizziert haben. Mein ganz besonderer Dank gilt meinen Betreuern, Prof. Dr. Volker Nollau und Dr. Peter Neumann, für Geduld, Unterstützung und unzählbar viele hilfreiche Diskussionen.

Contents

Chapter 1

Introduction

Let us imagine the following situation (cf. McMillan[11, p.51]): let N countries take part in world trade. Each country's sales depend not only on the price it decides to charge, but also on the prices charged by all the other countries. In addition, each country's demand curve fluctuates randomly. At the beginning of each period, and before the current realization of demand may be observed, each country decides its price and output for the duration of that period. Each country produces under constant returns to scale, with some average cost. How can each country, under the assumption of its being risk neutral, maximize the expected present value of its stream of rents over an infinite time horizon?

This example sketches a game-theoretic problem with ideas, relevant to economics, where countries play a game (i.e., compete in an international market) infinitely many times. From the mathematical point of view, the above game is a so-called *repeated game* with a random structure, also referred to as *stochastic game*: the game which is actually played in any particular time period is drawn according to a known probability distribution. This is just one example of multistage games which shall be the subject of this thesis.

In general, (noncooperative) Stochastic Games are dynamic, stochastic models of noncooperative competitive behaviour. They include as special cases

- static noncooperative games (games in the 'usual' sense)
- repeated games with complete information and
- Markov Decision Processes.

A remarkable fact is that the first paper on Stochastic Games appeared almost a decade before the development of the theory of Markov Decision Processes began. The term 'Stochastic Games' has been introduced by Lloyd S. Shapley in an article in 1953 (cf. Raghavan, Ferguson, Parthasarathy and Vrieze[22]), where he described a stochastic sequential decision problem with two decision makers, and used a value iteration algorithm to demonstrate the existence of a solution. It was only in the 1960's and 1970's that, almost independently from the latter, the theories of Markov Decision Processes and Dynamic Programming started evolving. As the subsequent research has shown, the theory of Markov Decision Processes provided the solution ideas for the theory of Stochastic games, but the first available algorithm for the latter also proved to be the basic method of solving the former. The works of Parthasarathy and Stern[17] (contains more than 150 references) and Raghavan et al.[22] provide a general overview of research in the field of Stochastic Games, containing among others Shapley's breaktrough article, but also recent developements in the fields of existence of various equilibria and algorithms for solving stochastic games.

In this chapter we shall describe the basic idea behind the theory of Stochastic Games. Throughout the chapters we shall use the notations and ideas widely used in static Game Theory. We shall assume that the terms, such as, matrix games, strategies, equilibria, etc. for static games, are already known and shall only use them as a tool for understanding and developing the concept of a Markov game. Some important results from Game Theory are listed in Appendix A. For further details we refer the reader to, e.g., Owen[16].

1.1 Shapley's model of a stochastic game

In the simplest case of finite static game-theoretical models, the game played is a two-person zero-sum game, often also called 'matrix game'. These games, for practical use represented by matrices, possess the following property (as stated by the celebrated Minimax Theorem of John von Neumann): *Given a real $u \times v$ matrix $C = (c_{ij})$, there exists a pair of probability vectors $\mathbf{x}^* = (x_1^*, \ldots, x_u^*)$ and $\mathbf{y}^* = (y_1^*, \ldots, y_v^*)$, such that for a unique constant ν*

$$\sum_{i=1}^{u} c_{ij} x_i^* \geq \nu \geq \sum_{j=1}^{v} c_{ij} y_j^* \quad \text{for all} \quad i, j.$$

We call $\nu := \text{val}\,[C]$ *the value* of the matrix game. The vectors $\mathbf{x}^*$ and $\mathbf{y}^*$ are called *optimal strategies*. The above game is called *zero-sum* since what one

player gets, the other player looses.

In his article, Shapley has posed the following question: what happens if the players play not just one matrix game, but different matrix games, one at each stage with a movement among these games depending on which entry was selected at the previous stage of the game?

Let $C^1, \ldots, C^m$ be real matrices known to the two players. By *state* s we mean the matrix game C^s. Players start in, say, state s. They play the matrix game C^s, player I thus receives some payoff from player II, and the game moves to $C^{s'}$ with a given probability that depends on the choices made by players I and II in state s. At the next stage they play $C^{s'}$ and so on.

The probability of transitions, known to both players, is assumed to be Markovian (hence the name Markov game), in the sense that the movement among games is determined only by the immediate past and not by the entire history. The aim of player I is to maximize his gain. The aim of player II is to minimize his loss.

Since such Markov game never ends, the overall payoff needs to be specified. We can study the solutions of such games by considering two types of overall payoff:

- β-discounted reward, for some discount factor β, and
- limiting average (undiscounted) reward.

In his fundamental paper, Shapley showed that the β-discounted finite stochastic games can be played optimally using stationary strategies, and provided a value iteration algorithm to find them.

Since then, many researchers tried to establish the existence of optimal strategies in both β-discounted and limiting average payoff cases. The known results for arbitrary state and action spaces and scalar payoffs so far are that, in general, optimal or at least ϵ-optimal solutions, exist in discounted stochastic games (see, e.g., Sobel[26], Nowak[15], etc.). In limiting average stochastic games the results are not so straightforward. In 1981, Mertens and Neyman[12] proved that zero-sum limiting average stochastic games have a value, but until now, even the existence of ϵ-equilibria in these games has been proven only for some subclasses of general limiting average stochastic games.

Studying stochastic games with vector-valued payoffs thus presents another level of difficulty and in this dissertation we shall limit ourselves only to

discounted Markov games, for which we shall prove the existence of stationary equilibria, generalize Shapley's result and develop algorithms for solving zero-sum Markov games.

1.2 Motivation for studying vector-valued payoffs

When making decisions in life, we often have to compare two or more non-commensurable objects, and the natural question arising from such problems is, how to characterize and find the optimal solutions. In the example from the beginning of the chapter, for example, the countries may be interested in trading many different products at once, such as electricity, petrol, etc. Intuitively, these otherwise noncommensurable objects can be compared by means of their value in some world currency. This already suggests an approach to vector optimization, which we shall use in Chapter 3.

The fundamentals of the theory of vector optimization reach back to 1970's and especially to the work of P.L. Yu[30] and [31]. Since then, many researchers contributed to the developement of the theory, e.g. R. Steuer[27], B. Rustem[24], etc.

In game-theoretic literature, great interest has also been shown in modelling and solving static games with vector payoffs (cf. H.W. Corley[3], Fernandez and Puerto[6], Ghose and Prasad[8]). The research in the field of dynamic game models has, however, been limited only to zero-sum stochastic games (K. Tanaka[28], L.A. Petrosjan and T. Tanaka[19]). By now, there are no known existing results for multi-player (non-zero-sum) noncooperative vector-valued stochastic games. We shall rely on the notations of K. Tanaka[28] and M.J. Sobel[26], develop a new calculus, and algorithms for solving such games.

The organization of the thesis is as follows: in Chapter 2 we shall develop a rigorous mathematical model for N-person non-zero-sum vector-valued Markov games. Chapter 3 explores the nature and properties of optimal solutions with respect to a given convex cone, and in Chapter 4, the dynamic programming approach is explored, and some solution procedures for two-person zero-sum games are provided.

Chapter 2

Model formulation

In this chapter we shall define the model for N-person noncooperative vector-valued Markov games and specify the notations used. We begin with a definition and explain the notations in the sections to follow.

Definition 2.1 *A **discounted noncooperative N-player Markov game with vector-valued reward** is the collection*

$$\left(N, S, A^1, \ldots, A^N, Q, \boldsymbol{r}^1, \ldots, \boldsymbol{r}^N, \beta\right) \tag{2.1}$$

where $N \in \mathbb{N}$ (number of players), S a non-empty, finite set (state space), $A^1, \ldots, A^N$ non-empty, finite sets (players' action spaces), $Q = \{q | q(\cdot|s, \boldsymbol{a}) : S \to [0,1], s \in S, \boldsymbol{a} \in A^1 \times \cdots \times A^N\}$ a family of probability measures on S which governs the law of transition from state to state, $\boldsymbol{r}^1, \ldots, \boldsymbol{r}^N$ p-dimensional vector-valued payoff functions to the players ($p \in \mathbb{N}$), and $0 \leq \beta < 1$ (given discount factor).

The game is played in stages. At each stage $t \in \mathbb{N}$, the game is in one of finitely many states; every player observes the current state s and chooses one of finitely many actions $a^i \in A^i, i = 1, \ldots, N$. The chosen multiaction $\boldsymbol{a} = (a^1, \ldots, a^N)$ at stage t, together with s, determines the payoff to the players at stage t, and the probability to select the next state.

2.1 State space

Let $\Gamma^1, \ldots, \Gamma^m$ be static N-person noncooperative games in Nash normal form with vector-valued reward functions to all players.

In a stochastic game, players choose games to play from the set $\{\Gamma^1, \ldots, \Gamma^m\}$. We identify this set with the set $S = \{1, 2, \ldots, m\}$ and call it **the state space**. Here m denotes the number of all possible games the players can play.

2.2 Action space

An **action space** is a set of all allowable actions a player can take in a certain state of the game. Let $s \in S$ be an arbitrary state. We denote the action space of player i in state s by

$$A_s^i, \quad card(A_s^i) = K_s^i < \infty$$

and write

$$A^i := \bigcup_{s \in S} A_s^i$$

for the action space of player i, and

$$A_s := A_s^1 \times \ldots \times A_s^N$$

for the action space of players $1, \ldots, N$ in the state s. We also assume that the following holds:

$$A_s^i \cap A_{s'}^i = \emptyset \text{ for } s \neq s', \forall i.$$

The action vector

$$\boldsymbol{a} = (a^1, a^2, \ldots, a^N) \in A := \prod_{i=1}^{N} A^i$$

shall be referred to as the **multiaction** of the players $1, \ldots, N$.

We shall also use the notations

$$\begin{aligned} K^i &:= |A^i| = \sum_{s \in S} K_s^i, \quad i = 1, \ldots, N \\ K_s &:= |A_s| = \prod_{i=1}^{N} K_s^i, \quad s \in S \\ K &:= |A| = \prod_{i=1}^{N} K^i. \end{aligned}$$

2.3 Rewards and transition probabilities

As a result of choosing a state $s \in S$ and a multiaction

$$\boldsymbol{a} = (a^1, \ldots, a^N) \in A,$$

the following occurs:

1. the players $i = 1, 2, \ldots, N$ receive **rewards**

   $$\mathbf{r}^i(s, \boldsymbol{a}) = (r_1^i(s, \boldsymbol{a}), \ldots, r_p^i(s, \boldsymbol{a})) \in \mathbb{R}^p,$$

   and

2. the state of the system at the next stage is determined by the probability vector

   $$\boldsymbol{q}(s, \boldsymbol{a}) = (q(1|s, \boldsymbol{a}), \ldots, q(m|s, \boldsymbol{a})) \in \Delta^m,$$

   where $q(\cdot|s, \boldsymbol{a}), (s, \boldsymbol{a}) \in S \times A$ is a given probability measure on S and $\Delta^m := \{\boldsymbol{u} = (u_1, \ldots, u_m) \in \mathbb{R}^m \,|\, \sum_{i=1}^m u_i = 1, u_i \geq 0 \,\forall i\}$.

This means: if the game is in state $s \in S$, and the players take the multiaction $\boldsymbol{a} \in A$ at stage t, it will enter the state $z \in S$ at stage $t + 1$ with probability

$$q(z|s, \boldsymbol{a}).$$

2.4 Actions, decision rules and strategies

A **(deterministic) decision rule** prescribes a procedure for action selection in each state at a specified time. That is, a decision rule is a function

$$d_t^i : S \to A^i, \quad i = 1, \ldots, N,$$

with $d_t^i(s) \in A_s^i$, which specifies the (deterministic) action choice of player i at time t.

Some important examples of decision rules are

- *history dependent* if they depend on the entire history of the system $h_t = (s_1, \boldsymbol{a}_1, s_2, \boldsymbol{a}_2, \ldots, s_{t-1}, \boldsymbol{a}_{t-1}, s_t) \in (S \times A)^{t-1} \times S$, where s_u and $\boldsymbol{a}_u$ denote the state and action at time $u = 1, \ldots, t$,

- *Markovian* if they depend on previous system states and actions only through the current state of the system,
- *deterministic* if they choose an action with certainty.

In a Markov game, we assume all decision rules to be Markovian and allow the players to randomize over their sets of actions in each game $\Gamma_s \equiv s \in S$. More precisely,

Definition 2.2 *A **randomized Markovian decision rule** of player i at time t in the game (2.1) is a function*

$$\sigma_t^i : S \to \mathbb{P}(A^i),$$

where $\mathbb{P}(A^i)$ is the set of probability measures on $(A^i, \mathcal{P}(A^i))$ and $\mathcal{P}(A^i)$ the power set of A^i. Thus, $\sigma_t^i(s)$ is a probability measure on A^i and $(\sigma_t^i(s))(a) := (\sigma_t^i(s))(\{a\})$ is the probability that the player i chooses the action $a \in A^i$ in state s at time t.

Furthermore, we shall refer to the vector of decision rules

$$\boldsymbol{\sigma}_t = (\sigma_t^1, \ldots, \sigma_t^N)$$

as the **multidecision rule** of the players at time t.

We defined our Markov game to be noncooperative, that is, all players choose their actions independently of other players. Therefore we shall require that the following holds

Definition 2.3

$$\boldsymbol{\sigma}_t(s, \boldsymbol{a}) := \prod_{i=1}^{N} (\boldsymbol{\sigma}_t^i(s))(a^i) \text{ for } \boldsymbol{a} = (a^1, \ldots, a^N).$$

Assumption. *If a multiaction $\boldsymbol{a} \in A$ is such that, for some $i \in \{1, \ldots, N\}$ and $s \in S$, $a^i \in A^i \backslash A_s^i$, we set*

$$(\rho(s))(\boldsymbol{a}) = 0$$

for all randomized Markovian decision rules ρ.

Every player's interest is to maximize his total income without making binding agreements. To achieve this goal, every player uses some **strategy**; this

is a plan that at any stage of play, given the current state, tells the player which randomized Markov decision rule to use.

More precisely, a strategy specifies the decision rule to be used at all decision epochs. It provides every player with a prescription for action selection under any possible history or future system state.

Definition 2.4 *A **(randomized Markov) strategy** σ^i of player i is a sequence of decision rules*

$$\sigma^i = (\sigma^i_1, \sigma^i_2, \dots),$$

where σ^i_t is a randomized Markovian decision rule of player i at time t.

Notation 2.5 *We let $\boldsymbol{\sigma}$ denote the **multistrategy** of N players $1, \dots, N$, where*

$$\boldsymbol{\sigma} = (\sigma^1, \dots, \sigma^N)$$

and σ^i is a strategy of player $i = 1, \dots, N$. Frequently we shall also use the notation

$$\boldsymbol{\sigma}^{-i} := (\sigma^1, \dots, \sigma^{i-1}, \sigma^{i+1}, \dots, \sigma^N)$$

for a multistrategy of $N-1$ players $1, \dots, i-1, i+1, \dots, N$, and shall denote by $(\rho, \boldsymbol{\sigma}^{-i})$ the multistrategy $\boldsymbol{\sigma}_\star$ of N players, such that

$$\begin{aligned} \sigma^j_\star &= \sigma^j \text{ for } j \neq i \\ \sigma^i_\star &= \rho. \end{aligned}$$

We call a strategy **stationary**, if only the current stage decides what randomized Markovian decision rule is to be used, and neither stage, nor history play a role, that is $\rho^i_t = \rho^i$ for all $t = 1, 2, \dots$. A stationary strategy of player i thus has the form

$$\sigma^i = (\rho^i, \rho^i, \dots) =: (\rho^i)^\infty.$$

Alternatively, a stationary strategy can be defined in the following way

Definition 2.6 *A **stationary strategy** σ^i for player $i, i = 1, \dots, N$ is an m-tuple of (probability) vectors, such that*

$$\begin{aligned} \sigma^i &= (\sigma^i_1, \dots, \sigma^i_m), \\ \sigma^i_s &= (\sigma^i_s(a^i_1), \dots, \sigma^i_s(a^i_{K^i_s})), \\ \sum_{k=1}^{K^i_s} \sigma^i_s(a^i_k) &= 1, \quad \sigma^i_s(a^i_k) \geq 0, \quad \forall k = 1, \dots, K^i_s, \quad \forall s = 1, \dots, m. \end{aligned}$$

Notation 2.7 *We let Π^i and Π^i_S denote the set of all (randomized Markov) strategies and all stationary randomized strategies of player $i, i = 1, \ldots, N$, respectively, and set*

$$\Pi := \Pi^1 \times \ldots \times \Pi^N$$

and

$$\Pi_S := \Pi^1_S \times \ldots \times \Pi^N_S.$$

With the assumption that the addition of two infinite vectors is performed componentwise, the following result is straightforward.

Proposition 2.8 *Π^i is a convex set for all $i = 1, \ldots, N$.*

Proof. Let σ and ρ be Markov strategies of player i for an arbitrary $i = 1, \ldots, N$ and let $\lambda \in [0, 1]$. The strategy $\lambda\sigma + (1-\lambda)\rho$ has components $\lambda\sigma_j + (1-\lambda)\rho_j, \quad j = 1, 2, \ldots$, where σ_j and ρ_j are randomized Markovian decision rules of player i. Therefore, for a fixed $s \in S$, the following holds

$$\begin{aligned} \sum_{a \in A^i} (\sigma_j(s))(a) &= 1, \quad (\sigma_j(s))(a) \geq 0 \quad \forall a \in A^i \\ \sum_{a \in A^i} (\rho_j(s))(a) &= 1, \quad (\rho_j(s))(a) \geq 0 \quad \forall a \in A^i. \end{aligned}$$

Then

$$\begin{aligned} & \sum_{a \in A^i} ((\lambda\sigma_j(s))(a) + ((1-\lambda)\rho_j(s))(a)) \\ = \; & \lambda \sum_{a \in A^i_s} (\sigma_j(s))(a) + (1-\lambda) \sum_{a \in A^i_s} (\rho_j(s))(a) \\ = \; & \lambda + (1-\lambda) = 1. \end{aligned}$$

Obviously, $\lambda(\sigma_j(s))(a) + (1-\lambda)(\rho_j(s))(a) \geq 0$ for all $a \in A^i$, therefore $\lambda\sigma_j + (1-\lambda)\rho_j \in \mathbb{P}(A^i)$ and is obviously Markovian for all $j = 1, 2, \ldots$. Thus we have shown that

$$\lambda\sigma + (1-\lambda)\rho \in \Pi^i$$

for arbitrary $\sigma, \rho \in \Pi^i$ and $0 \leq \lambda \leq 1$, therefore Π^i is a convex set. $\square$

2.5 The underlying stochastic process

Let S denote the state space and A the set of all multiactions of all players, and set

$$\Omega := (S \times A)^{\infty}.$$

A suitable σ-algebra is the power set of Ω,

$$\mathcal{F} = \mathcal{P}(\Omega) = \mathcal{P}((S \times A)^{\infty}).$$

Furthermore, let $p : s \mapsto p(s), s \in S$, denote the initial probability measure on S.

If a multistrategy $\boldsymbol{\sigma} \in \Pi$ is chosen, there exist a unique probability measure $\boldsymbol{P}_{\boldsymbol{\sigma}}$ (cf. Theorem of Ionescu-Tulcea [9, p. 149, Theorem A6]) on $(\Omega, \mathcal{F})$ and a stochastic process $(X_t, Y_t)_{t=1,2,\ldots}$ taking values in $(S \times A)$, such that

- $\boldsymbol{P}_{\boldsymbol{\sigma}}(X_1 = s) = p(s)$
- $\boldsymbol{P}_{\boldsymbol{\sigma}}(X_{t+1} = s | Z_t = (s_1, \boldsymbol{a}_1, \ldots, s_t), Y_t = \boldsymbol{a}) = q(s|s_t, \boldsymbol{a})$,
  if $\boldsymbol{P}_{\boldsymbol{\sigma}}(Z_t = (s_1, \boldsymbol{a}_1, \ldots, s_t), Y_t = \boldsymbol{a}) > 0$.
- $\boldsymbol{P}_{\boldsymbol{\sigma}}(Y_t = \boldsymbol{a} | Z_t = (s_1, \boldsymbol{a}_1, \ldots, s_t)) = \boldsymbol{\sigma}_t(s_t, \boldsymbol{a})$,
  if $\boldsymbol{P}_{\boldsymbol{\sigma}}(Z_t = (s_1, \boldsymbol{a}_1, \ldots, s_t)) > 0$.

Here,

$$X_t((s_1, \boldsymbol{a}_1, s_2, \boldsymbol{a}_2, \ldots)) := s_t$$

denotes the random state of the system at time t,

$$Y_t((s_1, \boldsymbol{a}_1, s_2, \boldsymbol{a}_2, \ldots))_{:} = \boldsymbol{a}_t$$

the random multiaction taken at time t and

$$Z_t((s_1, \boldsymbol{a}_1, s_2, \boldsymbol{a}_2, \ldots)) := (s_1, \boldsymbol{a}_1, \ldots, \boldsymbol{a}_{t-1}, s_t)$$

describes the random history at time t.

Equivalently, $\boldsymbol{P}_{\boldsymbol{\sigma}}$ can also be described as the unique probability measure on Ω for which

$$\boldsymbol{P}_{\boldsymbol{\sigma}}(Z_t = (s_1, \boldsymbol{a}_1, \ldots, s_t)) = p(s_1) \prod_{k=1}^{t-1} q(s_{k+1}|s_k, \boldsymbol{a}_k) \boldsymbol{\sigma}_k(s_k, \boldsymbol{a}_k). \tag{2.2}$$

2.6 The payoff structure

Let the players choose a multistrategy $\boldsymbol{\sigma}$ in the game (2.1). $\boldsymbol{\sigma}$ induces a discrete-time reward process, $\{(X_t, \boldsymbol{r}^1(X_t, Y_t), \ldots, \boldsymbol{r}^N(X_t, Y_t)) | t = 1, 2, \ldots\}$. The first component, X_t, represents the (random) state of the system at time t while the components $\boldsymbol{r}^1(X_t, Y_t), \ldots, \boldsymbol{r}^N(X_t, Y_t)$ represent the rewards, received by the players $1, \ldots, N$, when they use the multiaction Y_t in the state X_t. Here,

$$\boldsymbol{r}^i : (S \times A) \to \mathbb{R}^p, \quad i = 1, \ldots, N$$

are vector-valued functions on $S \times A$. Thus

$$\boldsymbol{R}_t^i := \boldsymbol{r}^i(X_t, Y_t), \quad i = 1, \ldots, N,$$

represents a random vector denoting the direct payoff to player i at the stage t of the game. Furthermore, let

$$\boldsymbol{R}^i : \Omega \to \mathbb{R}^p$$

denote the overall random discounted vector-valued payoff to player i. If by using the multistrategy $\boldsymbol{\sigma}$ the sequence $\omega = (s_1, \boldsymbol{a}_1, s_2, \boldsymbol{a}_2, \ldots) \in \Omega$ occurs, then player i receives the discounted reward

$$\boldsymbol{R}^i(\omega) = \sum_{t=1}^{\infty} \beta^{t-1} \boldsymbol{R}_t^i(\omega) = \sum_{t=1}^{\infty} \beta^{t-1} \boldsymbol{r}^i(s_t, \boldsymbol{a}_t).$$

Thus the expected vector-valued reward to player i at each time t is given by

$$\begin{aligned}
E_{\boldsymbol{\sigma}}\left[\boldsymbol{R}_t^i\right] &= E_{\boldsymbol{\sigma}}\left[\boldsymbol{r}^i(X_t, Y_t)\right] \\
&= \sum_{s \in S} \left(\sum_{\boldsymbol{a} \in A} \boldsymbol{r}^i(s, \boldsymbol{a}) \boldsymbol{P}_{\boldsymbol{\sigma}}(Y_t = \boldsymbol{a} | X_t = s) \right) \boldsymbol{P}_{\boldsymbol{\sigma}}(X_t = s) \\
&= \sum_{\substack{s \in S \\ \boldsymbol{a} \in A}} \boldsymbol{r}^i(s, \boldsymbol{a}) \boldsymbol{P}_{\boldsymbol{\sigma}}(Y_t = \boldsymbol{a}, X_t = s) \\
&= \left(\sum_{\substack{s \in S \\ \boldsymbol{a} \in A}} r_k^i(s, \boldsymbol{a}) \boldsymbol{P}_{\boldsymbol{\sigma}}(Y_t = \boldsymbol{a}, X_t = s) \right)_{k=1}^{p}
\end{aligned}$$

where $E_{\boldsymbol{\sigma}}$ denotes the expectation with respect to players' using the multistrategy $\boldsymbol{\sigma}$.

With given discount factor β and multistrategy $\boldsymbol{\sigma}$ the expected discounted vector-valued reward for the player i is defined to be

$$f^i(\boldsymbol{\sigma}) := \lim_{T\to\infty} E_{\boldsymbol{\sigma}}\left[\sum_{t=1}^{T}\beta^{t-1}\boldsymbol{R}_t^i\right] \tag{2.3}$$

for $0 \le \beta < 1$. The above limit exists, when

$$\max_{s\in S}\max_{\boldsymbol{a}\in A_s}\max_{1\le k\le p}|r_k^i(s,\boldsymbol{a})| = M < \infty.$$

In this case, we write

$$\begin{aligned} f^i(\boldsymbol{\sigma}) &= E_{\boldsymbol{\sigma}}\left[\boldsymbol{R}^i\right] = E_{\boldsymbol{\sigma}}\left[\sum_{t=1}^{\infty}\beta^{t-1}\boldsymbol{r}^i(X_t,Y_t)\right] \\ &= \left(\sum_{t=1}^{\infty}\beta^{t-1}\sum_{\substack{s\in S\\ \boldsymbol{a}\in A}} r_k^i(s,\boldsymbol{a})\boldsymbol{P}_{\boldsymbol{\sigma}}(Y_t=\boldsymbol{a},X_t=s)\right)_{k=1}^{p} \\ &= \left(f_1^i(\boldsymbol{\sigma}),\ldots,f_p^i(\boldsymbol{\sigma})\right). \end{aligned}$$

Notation 2.9 *We shall often use the following notation*

$$f^i(\rho^i,\boldsymbol{\sigma}^{-i}) := f^i((\rho^i,\boldsymbol{\sigma}^{-i}))$$

and define the notation $f^i(\boldsymbol{\sigma})(s)$ *to be*

$$f^i(\boldsymbol{\sigma})(s) := E_{\boldsymbol{\sigma}}\left[\sum_{t=1}^{\infty}\beta^{t-1}\boldsymbol{r}^i(X_t,Y_t)\Big|X_1=s\right],$$

that is, payoff to player i *by using the multistrategy* $\boldsymbol{\sigma}$ *conditioned on the starting state* s.

Chapter 3

Convex cones and D-equilibria

In multi-criteria optimization one encounters many problems that do not arise in the corresponding scalar optimization problems. Most importantly, in seeking *better* or *the best* sloution one is confined to the use of a partial-order relation, which eliminates the possibilities of the existence of a unique solution. One can only speak about *nondominated* solutions with respect to the selected relation. This feature strongly affects the subsequent development of algorithms for such problems.

The most famous partial-order relation is that of *Pareto*, but as its generalization, the most widespread and well-researched concept in vector optimization theory is the concept of relations, induced by convex cones, as discussed in the fundamental article by Yu[30]. In his book a decade later, Yu[31] researches the aspects of preferences and cone convexity together with multi-criteria optimization even more thoroughly.

The results of this theory can also be applied to general vector-valued Markov games as we shall demonstrate in this chapter. We shall extend the results of Tanaka[28] on N-player Markov games, define the concept of optimality of vector-valued payoff functions over convex cones, state the different notions of optimality in stochastic games and examine the existence and properties of D-equilibria in Markov games.

All the notations are (if not otherwise stated) in accordance with previously defined symbols.

3.1 Preference relations and optimality

In this section we shall state some definitions which shall be of use later in defining the concept of optimality.

Definition 3.1 *Let $x := (x_1, \ldots, x_p)$ and $y := (y_1, \ldots, y_p)$ be two vectors in $\mathbb{R}^p$ with **inner product** given by*

$$\langle x, y \rangle := \sum_{i=1}^{p} x_i y_i.$$

Let $\mathbf{0} := (0, 0, \ldots, 0) \in \mathbb{R}^p$.

Definition 3.2 *A set $C \subset \mathbb{R}^p$ is a **cone** if $\lambda C \subset C$ for all $\lambda \in \mathbb{R}_+$. The **positive polar cone** of C is given by*

$$C^* = \{y \in \mathbb{R}^p : \langle x, y \rangle \geq 0 \quad \forall x \in C\}.$$

*A **cone spanned by a set** $E \subset \mathbb{R}^p$ is defined by*

$$[E] = \{y \in \mathbb{R}^p : y = \lambda x, \quad x \in E, \quad \lambda \in \mathbb{R}_+\}.$$

*A cone C is **pointed** if $\mathbf{0} \in C$.*

*A set $C \in \mathbb{R}^p$ is a **convex cone** if it is a convex set and a cone at the same time.*

Definition 3.3 *Let D be a convex cone in $\mathbb{R}^p$. We say that a set $A \in \mathbb{R}^p$ is D-**convex** (respectively $-D$-**convex**) if $A + D$ (respectively $A - D$) is a convex set in $\mathbb{R}^p$.*

Now, let $D \subset \mathbb{R}^p$ be a given convex cone. In a Markov game, the set of all possible alternatives of a player, say i, is the set of all Markov strategies, Π, and the totality of each possible outcome, the so-called **outcome space**, is $Y^i := \{f^i(\boldsymbol{\sigma}) | \boldsymbol{\sigma} \in \Pi\} \subset \mathbb{R}^p$. It is evident (see, e.g., Yu[30]), that every convex cone induces a preference relation on the outcome space, namely:
Let $y_1, y_2 \in Y^i$ for some $i = 1, \ldots, N$. The relation

$$y_1 \succ y_2 \iff y_1 - y_2 \in D \backslash \{\mathbf{0}\},$$

(that is, y_1 'is preferred to' y_2) defines a partial order on Y^i.

Example 3.4 (Pareto optimality) *Let*

$$D = \{x \in \mathbb{R}^p | x_i \geq 0, i = 1, \ldots, p\}.$$

This is the so-called ***Pareto cone*** *and the relation it induces is the following:*

$$\begin{array}{lcll} x \overset{\geq}{=} y & \iff & x_i \geq y_i & i = 1, \ldots, p \\ x \geq y & \iff & x_i \geq y_i, & i = 1, \ldots, p \text{ and } x \neq y \\ x > y & \iff & x_i > y_i, & i = 1, \ldots, p \end{array}$$

for x and y in $\mathbb{R}^p$. In this context, the relation $\not\geq$ is defined by

$$x \not\geq y \iff \neg x > y$$

Of course, in general, $\not\geq$ is not a partial-order relation, but for $p = 1$, it is equivalent to the relation $\leq$ in $\mathbb{R}$, which is a partial-order relation.

3.2 D-equilibria

The traditional way of solving problems, proposed by Game Theory, lies in finding the so-called *equilibria*, that is, those elements of the players' strategy sets that yield the optimal payoff. In this section, we shall precisely define what we understand under the word 'equilibrium'; however, there is no unified approach found in the literature. The most widely used is the concept of *Nash equilibria*, although especially in the theory of two-player zero-sum scalar-valued stochastic games extensive research has been done using *sequential Stackelberg equilibria*, and in non-zero-sum games the concept of *correlated equilibria* (see, e.g., Raghavan et. al.[22]).

In this section, we shall extend the notion of Nash equilibria to N-player non-zero-sum vector-valued Markov games.

We shall start with some general definitions.

Notation 3.5 *In the following, let the sets $L \subset \mathbb{R}^p$ and $D := L \cup \{\mathbf{0}\}$ fulfill the following properties*

1. *D is a pointed convex cone*
2. *$L^\bullet := \{y \in \mathbb{R}^p : \langle x, y \rangle > 0 \quad \forall x \in L\} \neq \emptyset$.*

Remark 3.6 *It follows from the above notation that L is a convex cone without $\mathbf{0} \in \mathbb{R}^p$.*

Definition 3.7 *Let D be a convex cone as defined above. A strategy σ_*^i of player i is* ***D-optimal*** *(D-efficient) for the multistrategy $\boldsymbol{\sigma}^{-i} := (\sigma^1, \ldots, \sigma^{i-1}, \sigma^{i+1}, \ldots, \sigma^N)$ of players $1, \ldots, i-1, i+1, \ldots, N$, if there exists no other strategy $\rho \in \Pi^i$, such that*

$$f^i(\sigma_*^i, \boldsymbol{\sigma}^{-i}) \in f^i(\rho, \boldsymbol{\sigma}^{-i}) - L, \tag{3.1}$$

where $L \subset D$ is as defined above.

Example 3.8 *In the one-dimensional case, that is $f^i : \Pi \to \mathbb{R}$ for all $i = 1, \ldots, N$, L can be interpreted as $L = (0, \infty)$ and the relation (3.1) is as follows:*
The strategy σ_*^i is D-optimal for player i for the strategy $\boldsymbol{\sigma}^{-i}$ if there exists no other strategy $\rho \in \Pi^i$ such that

$$f^i(\sigma_*^i, \boldsymbol{\sigma}^{-i}) \in f^i(\rho, \boldsymbol{\sigma}^{-i}) - (0, \infty)\,.$$

In other words, there exists no such strategy ρ that

$$f^i(\sigma_*^i, \boldsymbol{\sigma}^{-i}) < f^i(\rho, \boldsymbol{\sigma}^{-i}) \quad i = 1, \ldots, N,$$

which is exactly the definition of optimality in the scalar case.

Example 3.9 (Pareto Optima) *If $p = 2$, that is $f^i : \Pi \to \mathbb{R}^2$ for all $i = 1, \ldots, N$, and D is a Pareto cone, then σ_*^i of player i is Pareto-optimal for $\boldsymbol{\sigma}^{-i}$ if*

$$f^i(\sigma_*^i, \boldsymbol{\sigma}^{-i}) \overset{\geq}{=} f^i(\rho, \boldsymbol{\sigma}^{-i}) \quad \forall \rho \in \Pi^i.$$

Example 3.10 (Bimatrix game) *Let D be a Pareto cone, $N = 2$, $S = \{1\}$, $f^i : \Pi \to \mathbb{R}^p$ for $i = 1, 2$ and let there be only one stage of the game[1], that is, $t = 1$. Then, the game in question is a static bimatrix game with equilibrium point $\boldsymbol{\sigma}_* = (\sigma_*^1, \sigma_*^2)$ for which*

$$f^i(\boldsymbol{\sigma}_*) \overset{\geq}{=} f^i(\rho^i, \sigma_*^{-i}) \quad i = 1, 2,\ \rho^i \in \Pi^i.$$

Of course, in this case, Π^i is the set of all mixed strategies (in the sense of static Game Theory) of player i in this bimatrix game, i.e. the set of all probability distributions over his set of actions in the game.

[1] In a game with infinite horizon this can be interpreted as a two-state game where we start in state 1 with probability one, jump to state 2 in stage 2, for which the payoff is 0, and stay in state 2 with probability 1 on each following stage.

The above definitions lead to the natural conclusion,

Definition 3.11 *A multistrategy $\boldsymbol{\sigma}_*$ of N players is a **D-equilibrium** if for each player i, σ_*^i is D-optimal for the multistrategy $\boldsymbol{\sigma}_*^{-i}$ of $N-1$ players $1,\dots,i-1,i+1,\dots,N$.*

Remark 3.12 *Analogously with Yu[30] we can define*

$$G^i(\boldsymbol{\sigma}^{-i}) = \{f^i(\rho,\boldsymbol{\sigma}^{-i})|\rho\in\Pi^i\}$$

and denote by

$$Ext[G^i(\boldsymbol{\sigma}^{-i})|D] = \{f^i(\rho,\sigma^{-i})|\rho\in\Pi^i \text{ D-optimal for player } i \text{ for } \boldsymbol{\sigma}^{-i}\}\subset\mathbb{R}^p$$

the set of all expected discounted vector valued rewards to player i which correspond to all D-optimal strategies of player i for a given multistrategy $\boldsymbol{\sigma}^{-i}$ of $N-1$ players $1,\dots,i-1,i+1,\dots,N$. With this notation, the latter definition suggests, that a multistrategy $\boldsymbol{\sigma}_*$ of N players is a D-equilibrium iff

$$f^i(\boldsymbol{\sigma}_*)\in Ext[G^i(\boldsymbol{\sigma}_*^{-i})|D] \quad \forall i=1,\dots,N. \tag{3.2}$$

Remark 3.13 *If $L=Int(D)$ (respectively $L=D-\{0\}$) for a given convex cone $D\in\mathbb{R}^p$, the strategy σ_*^i for which (3.1) holds, is said to be D-**weak optimal** (respectively D-**strong optimal**). Hence a multistrategy $\boldsymbol{\sigma}_*$ in (3.2) is said to be a D-**weak equilibrium** (respectively D-**strong equilibrium**).*

A common technique of solving vector optimization problems is finding the solution of the corresponding scalar-valued problem. For this we need the following

Definition 3.14 *A multistrategy $\boldsymbol{\sigma}_*$ of N players is a **weight D-equilibrium** with respect to the weight vectors $d^1,\dots,d^N\in\mathbb{R}^p$ if, for each player $i=1,\dots,N$, the following holds*

1. $d^i\in L^\bullet$
2. $\langle d^i, f^i(\rho,\boldsymbol{\sigma}_*^{-i})\rangle \le \langle d^i, f^i(\boldsymbol{\sigma}_*)\rangle$ *for each strategy ρ of player i.*

This definition corresponds to the definition of Nash equilibria for games with scalar-valued functions. The following result is straightforward.

Lemma 3.15 *Let $D = L \cup \{\mathbf{0}\}$ be a given convex cone (cf. Notation 3.5), $d^1, \ldots, d^N \in L^\bullet$ and let $\boldsymbol{\sigma}_*$ be a weight D-equilibrium with respect to $d^1, \ldots, d^N$ in the game (2.1), that is*

$$\langle d^i, f^i(\sigma^i, \boldsymbol{\sigma}_*^{-i})\rangle \leq \langle d^i, f^i(\boldsymbol{\sigma}_*)\rangle \tag{3.3}$$

for $d^i \in L^\bullet$ and all $\sigma^i \in \Pi^i$, $i = 1, \ldots, n$. Then $\boldsymbol{\sigma}_*$ is a D-equilibrium.

Proof. Let $d^1, \ldots, d^N \in L^\bullet$, let $\boldsymbol{\sigma}_*$ be a weight D-equilibrium with respect to $d^1, \ldots, d^N$ and suppose that $\boldsymbol{\sigma}_*$ is not a D-equilibrium. Then there exists a $j \in \{1, \ldots, N\}$ and a strategy $\sigma^j \in \Pi^j$, such that

$$f^j(\boldsymbol{\sigma}_*) \in f^j(\sigma^j, \boldsymbol{\sigma}_*^{-j}) - L$$

This relation claims that there exists a $d_* \in L$, such that

$$f^j(\boldsymbol{\sigma}_*) = f^j(\sigma^j, \boldsymbol{\sigma}_*^{-j}) - d_*. \tag{3.4}$$

Now take the inner product with $d^j \in L^\bullet$ in (3.4) to obtain

$$\langle d^j, f^j(\boldsymbol{\sigma}_*)\rangle = \langle d^j, f^j(\sigma^j, \boldsymbol{\sigma}_*^{-j}) - d_*\rangle < \langle d^j, f^j(\sigma^j, \boldsymbol{\sigma}_*^{-j})\rangle,$$

which contradicts the weight D-optimality of $\boldsymbol{\sigma}_*$. □

This lemma confirms a well-known fact from the theory of vector optimization that it is always possible to find a solution of a vector-valued optimization problem through its scalarized version by using some weight vectors for the payoff functions.

It is also well known, cf. e.g., Pascoletti, Serafini[18], that vector- and scalar-valued optimization problems are 'only' almost equivalent. In other words,

Theorem 3.16 *Let $D = L \cup \{\mathbf{0}\}$ be a given convex cone, where L is an open set in $\mathbb{R}^p$ and $\boldsymbol{\sigma}_*$ a D-equilibrium in the game (2.1). Furthermore, let $G^i(\boldsymbol{\sigma}_*^{-i})$ be $-L$-convex for all $i = 1, \ldots, N$. Then there exist weight vectors $d^1, \ldots, d^N \in L^\bullet$, such that $\boldsymbol{\sigma}_*$ is a weight D-equilibrium with respect to $d^1, \ldots, d^N$ in the so-called scalarized game with payoff functions $\langle d^1, \boldsymbol{r}^1\rangle, \ldots, \langle d^N, \boldsymbol{r}^N\rangle$.*

Proof. Saying that $\boldsymbol{\sigma}_*$ is a D-equilibrium is equivalent to the following relation

$$f^i(\boldsymbol{\sigma}_*) \notin G^i(\boldsymbol{\sigma}_*^{-i}) - L \quad \forall i = 1, \ldots, N. \tag{3.5}$$

Since we assumed that $G^i(\boldsymbol{\sigma}_*^{-i})$ are $-L$-convex for all $i = 1, \ldots, N$, the sets $G^i(\boldsymbol{\sigma}_*^{-i}) - L$ are also convex. Now we may use the Large Separation Theorem (Aubin[1, p.32, Theorem 2.5]) to obtain for each i a $d^i \in \mathbb{R}^p, d^i \neq \mathbf{0}$, such that

$$\begin{aligned} \langle d^i, f^i(\boldsymbol{\sigma}_*)\rangle &\geq \sup_{x \in G^i(\boldsymbol{\sigma}_*^{-i}) - L} \langle d^i, x\rangle \\ &= \sup_{\substack{\rho \in \Pi^i \\ y \in -L}} \langle d^i, f^i(\rho, \boldsymbol{\sigma}_*^{-i}) + y\rangle \\ &= \sup_{\rho \in \Pi^i} \langle d^i, f^i(\rho, \boldsymbol{\sigma}_*^{-i})\rangle + \sup_{y \in -L} \langle d^i, y\rangle. \end{aligned}$$

Hence, $\sup_{y \in -L} \langle d^i, y\rangle \leq 0$, and thus

$$\langle d^i, y\rangle \geq 0 \quad \forall y \in L,$$

that is, $d^i \in L^*, i = 1, \ldots, N$. Noting that $d^i \neq \mathbf{0}$ and, because L is open, $L^\bullet = L^* \backslash \{\mathbf{0}\}$ (cf. K. Tanaka[28]), we obtain

$$d^i \in L^\bullet \quad \forall i = 1, \ldots, N,$$

therefore d^i satisfy the relation (3.3) for all $i = 1, \ldots, N$ and a given multi-strategy $\boldsymbol{\sigma}_*$ and so we have proved that $\boldsymbol{\sigma}_*$ is a weight D-equilibrium with respect to the weight vectors $d^1, \ldots, d^N$. $\square$

It is perhaps somewhat difficult to imagine the sufficient conditions from the above theorem at the first sight. The following corollary states them for a special case of Pareto-optima.

Corollary 3.17 *Let D be a Pareto cone, satisfying the conditions of Theorem 3.16 and $\boldsymbol{\sigma}_*$ a D-equilibrium in the game (2.1). If the functions $g^i := f^i(\cdot, \boldsymbol{\sigma}_*^{-i}) : \Pi^i \to \mathbb{R}^p$ are concave for all $i = 1, \ldots, N$, then there exist weight vectors $d^1, \ldots, d^N \in L^\bullet$, such that $\boldsymbol{\sigma}_*$ is a weight D-equilibrium in the scalarized game with respect to $d^1, \ldots, d^N$.*

Proof. Let D be a Pareto cone, satisfying the conditions of the theorem, that is, $L = D\backslash\{\mathbf{0}\} = \{x \in \mathbb{R}^p | x > \mathbf{0}\}$, and let $\boldsymbol{\sigma}_*$ be a D-equilibrium in the game (2.1).

We have already seen (cf. Proposition 2.8), that Π^i are convex sets for $i = 1, \ldots, N$. Since the functions g^i are concave, we may use a result by Yu[31, p. 26, Theorem 3.7] to obtain that the sets

$$G^i(\boldsymbol{\sigma}_*^{-i}) = \{f^i(\sigma, \boldsymbol{\sigma}_*^{-i}) | \sigma \in \Pi^i\}$$

are $-L$-convex, where $L = D\backslash\{\mathbf{0}\}$. The result then follows directly from Theorem 3.16. □

Remark 3.18 *A function $f : \Pi \to \mathbb{R}^p$ is concave if, and only if,*

$$\lambda f(x) + (1-\lambda) f(y) \overset{\leq}{=} f(\lambda x + (1-\lambda) y)$$

for any $\lambda \in [0,1]$ and any $x, y \in \Pi$.

3.3 The existence of D-equilibria

The existence of stationary equilibria, that is, equlibria in stationary strategies, for scalar-valued stochastic games has been widely researched in literature. Known results at present are, that stationary equilibria exist in several classes of stochastic games (cf. Sobel[26], Mertens and Neyman[12], etc.). The usual approach is the research of stationary equilibria, because by using them, the players face a Markov decision process at each time $t \in \mathbb{N}$, which simplifies the already complex algorithms for computation of such equilibria.

We have already shown that there exists a close relationship between vector- and scalar-valued stochastic games, and now we shall prove that stationary equilibria in our game model exist.

Theorem 3.19 *There exists a stationary D-equilibrium in the game (2.1).*

Proof. Let $d^1, \ldots, d^N \in L^\bullet$ be arbitrary fixed weight vectors. For each player $i = 1, \ldots, N$ and each multistrategy $\boldsymbol{\sigma}$ we can rewrite the scalar valued reward function $< d^i, f^i(\boldsymbol{\sigma}) >$ as

$$\begin{aligned} < d^i, f^i(\boldsymbol{\sigma}) > &= \sum_{k=1}^{p} d^i_k f^i_k(\boldsymbol{\sigma}) \\ &= \sum_{t=1}^{\infty} \beta^{t-1} \sum_{k=1}^{p} d^i_k E_{\boldsymbol{\sigma}}[(\boldsymbol{R}^i_t)_k] \\ &= \sum_{t=1}^{\infty} \beta^{t-1} E_{\boldsymbol{\sigma}}[< d^i, \boldsymbol{R}^i_t >], \end{aligned}$$

where $(\boldsymbol{R}^i_t)_k$ is the k-th component of the random reward vector $\boldsymbol{R}^i_t$. Thus we have modified the original game system to a scalar valued N-person stochastic

game

$$(N, S, A^1, \ldots, A^N, Q, \langle d^1, \boldsymbol{r}^1 \rangle, \ldots, \langle d^N, \boldsymbol{r}^N \rangle, \beta). \tag{3.6}$$

This is a finite stochastic game, hence by Sobel[26] there exists a stationary equilibrium in this game. The assertion then follows directly from Lemma 3.15. □

3.4 D-equilibria and subgradient

Searching an equilibrium in a game is equivalent to optimizing the players' criterion functions, in our case the expected discounted vector-valued reward functions.

The issue of function optimization is at the very core of nonlinear analysis. Provided a function is, for example, convex and lower semi-continuous on a suitable space, its minimum (equivalently, its maximum) can be found by computing the sub-differential of its support function. We shall prove in this section that this remains valid for our reward functions, too.

Definition 3.20 *Let $\boldsymbol{\sigma}$ be a multistrategy of N players in the game (2.1). The support function $U_i(\boldsymbol{\sigma}^{-i})$ of $f^i(\boldsymbol{\sigma})$ is defined by*

$$U_i(\boldsymbol{\sigma}^{-i})(d) = \sup_{\rho \in \Pi^i} \langle d, f^i(\rho, \boldsymbol{\sigma}^{-i}) \rangle, \quad d \in \mathbb{R}^p$$

We have seen that under certain conditions a D-equilibrium in a stochastic game is also a weight D-equilibrium for some weight vectors $d^1, \ldots, d^N \in L^\bullet$. This gives us the following

Definition 3.21 *We call weight vectors $d^1, \ldots, d^N \in L^\bullet$ **D-multipliers** of a multistrategy $\boldsymbol{\sigma} \in \Pi$ iff $\boldsymbol{\sigma}$ is a weight D-equilibrium with respect to $d^1, \ldots, d^N$ in the Markov game (2.1).*

Another definition we shall use is the following

Definition 3.22 *A function $g : E \subseteq \mathbb{R}^p \to \mathbb{R}$ is called **sub-differentiable** at $x_0 \in E$ if there exists a vector $y \in \mathbb{R}^p$, such that*

$$g(x) - g(x_0) \geq \langle x - x_0, y \rangle \quad \forall x \in E.$$

*Such a vector y is a **subgradient** of g at x_0. The set of all subgradients of g at x_0 shall be denoted by $\underline{\partial} g(x_0)$*

Example 3.23 *Let $g(x) = |x|$ for $x \in \mathbb{R}^p$ and $x_0 = \mathbf{0}$. According to the above definition, a vector $y \in \mathbb{R}^p$ is a subgradient of g at $\mathbf{0}$ if*

$$g(x) - g(\mathbf{0}) \geq \langle x - \mathbf{0}, y \rangle,$$

that is,

$$|x| \geq \langle x, y \rangle.$$

In the special case of $p = 1$ we obtain the following:

$y \in \mathbb{R}$ is a subgradient of g at 0 iff $|x| \geq xy$. Equivalently,

$$-x \leq xy \leq x \quad \forall x \in \mathbb{R}$$

or rather

$$-1 \leq y \leq 1.$$

Thus we have obtained that each $y \in [-1, 1]$ is a subgradient of $g(x) = |x|$ at $\mathbf{0}$.

It is a classical result of optimization theory that the optima of a function can be obtained by sub-differentiating its support function (as defined above). To state this more precisely, we shall prove the following theorem, a generalization of Theorem 5.1 in Tanaka[28, p. 307].

Theorem 3.24 *A multistrategy $\boldsymbol{\sigma}_*$ is a D-equilibrium with respect to D-multipliers $d^1, \ldots, d^N \in L^\bullet$ in the game (2.1) if and only if $f^i(\boldsymbol{\sigma}_*)$ is a subgradient of the support function $U_i(\boldsymbol{\sigma}_*^{-i})$ at d^i for all $i = 1, \ldots, N$, that is*

$$f^i(\boldsymbol{\sigma}_*) \in \underline{\partial} U_i \left(\boldsymbol{\sigma}_*^{-i}\right)(d^i) \quad i = 1, \ldots, N.$$

Proof. Sufficiency: If $f^i(\boldsymbol{\sigma}_*)$ is a subgradient of $U_i(\boldsymbol{\sigma}_*^{-i})$ at d^i for all $i = 1, \ldots, N$, then for all $d \in L^\bullet \cup \{\mathbf{0}\}$ the following holds

$$U_i(\boldsymbol{\sigma}_*^{-i})(d) - U_i(\boldsymbol{\sigma}_*^{-i})(d^i) \geq \langle d - d^i, f^i(\boldsymbol{\sigma}_*) \rangle.$$

In particular, for $d = \mathbf{0}$,

$$U_i(\boldsymbol{\sigma}_*^{-i})(d^i) \leq \langle d^i, f^i(\boldsymbol{\sigma}_*) \rangle,$$

that is

$$\langle d^i, f^i(\rho, \boldsymbol{\sigma}_*^{-i}) \rangle \leq \langle d^i, f^i(\boldsymbol{\sigma}_*) \rangle \tag{3.7}$$

for all $\rho \in \Pi^i$ and all $i = 1, \ldots, N$.

(3.7) implies that $\boldsymbol{\sigma}_*$ is a weight D-equilibrium with respect to the weight vectors
$d^1, \ldots, d^N$and thus (cf. Lemma 3.15) also a D-equilibrium.

Necessity: If $\boldsymbol{\sigma}_*$ is a D-equilibrium associated with the D-multiplier $d^1, \ldots, d^N \in L^\bullet$, then

$$U_i(\boldsymbol{\sigma}_*^{-i})(d^i) = \langle d^i, f^i(\boldsymbol{\sigma}_*)\rangle \quad \forall i = 1, \ldots, N. \tag{3.8}$$

By definition of the support function the following holds

$$U_i(\boldsymbol{\sigma}_*^{-i})(d) \geq \langle d, f^i(\boldsymbol{\sigma}_*)\rangle \tag{3.9}$$

for all $d \in L^\bullet \cup \{\mathbf{0}\}$, $i = 1, \ldots, N$.

By subtracting (3.8) from (3.9) we obtain

$$U_i(\boldsymbol{\sigma}_*^{-i})(d) - U_i(\boldsymbol{\sigma}_*^{-i})(d^i) \geq \langle d - d^i, f^i(\boldsymbol{\sigma}_*)\rangle$$

for all $i = 1, \ldots, N$ and all $d \in L^\bullet \cup \{\mathbf{0}\}$. Thus

$$f^i(\boldsymbol{\sigma}_*) \in \underline{\partial} U_i(\boldsymbol{\sigma}_*^{-i})(d^i) \quad \forall i = 1, \ldots, N,$$

which proves the theorem. □

The above theorem suggests the following, for an N-player Markov game with the payoff functions $f^1, \ldots, f^N$: if we construct a support function for each of the functions f^i, and compute for each at least one of their subgradients, this 'joint' subgradient is a D-equilibrium, together with some implicitely obtained weight vectors.

Equivalently, the problem posed is a mathematical program with nonlinear objective (searching a minimum of a nonlinear function). Thus, for solving, this theorem suggests the nonlinear programming approach, for scalar-valued stochastic games already developed by Filar, Schultz, Thuijsman and Vrieze[7].

Chapter 4

Solution procedures for two-person zero-sum games

In researching possible solution procedures for Markov games we shall concentrate on their special subclass, two-person zero-sum games. Partly, because the existing theory for scalar-valued two-person zero-sum stochastic games is well-developed, but mostly because of the relative simplicity of the model one can straightforwardly apply the results from the theory of Markov Decision Processes to obtain value and policy iteration algorithms. An overall review and an analysis of existing algorithms for scalar-valued stochastic games is to be found in Breton et al.[2].

Up to the present point there exist two possible ways of devising algorithms for scalar stochastic games, the dynamic programming method and nonlinear programming. The applications of the former have most thoroughly been summed up in Breton et al.[2], where the authors also provided a nonlinear algorithm for solving general N-person stochastic games. In 1991, Filar et al.[7] developed a general nonlinear algorithm for N-person stochastic games. Because of the general complexity of the model, even with scalar-valued games, the algorithms often prove to be troublesome in that they albeit converge, but the criterion function as such is often very ill-conditioned, as seen in Breton et al.[2]

No algorithms, however, yet exist for vector-valued stochastic games. In this chapter we shall explore the possible solution procedures and devise algorithms when possible.

4.1 The dynamic programming formalism

As in Markov Decision Process Theory, one can use the Denardo dynamic programming formalism for zero-sum stochastic games, too.

Let

$$V := \{v : S \to \mathbb{R}^p\}$$

be the space of all (bounded) p-dimensional vector-valued functions on S and denote by

$$h : S \times A \times V \to \mathbb{R}^p$$

the local income function, given by

$$h(s, \boldsymbol{a}, v) := \boldsymbol{r}(s, \boldsymbol{a}) + \beta \sum_{s' \in S} q(s'|s, \boldsymbol{a}) v(s'), \quad \beta \in [0, 1). \tag{4.1}$$

For a fixed bidecision rule δ we denote

$$h(s, \delta(s), v) := \sum_{\boldsymbol{a} \in A} h(s, \boldsymbol{a}, v)(\delta(s))(\boldsymbol{a}).$$

Thus the dynamic programming operator $H_\delta : V \to V$ is given by

$$[H_\delta v](s) := h(s, \delta(s), v). \tag{4.2}$$

Define by

$$\| v \| := \max_{s \in S} \max_{1 \le k \le p} |v_k(s)| \tag{4.3}$$

for $v \in V$ the norm on V.

As it has been shown in Chapter 3, we may search the equilibria of Markov games only in the class Π_S of stationary strategies. This shall simplify the subsequent algorithms, as well as provide some interesting theoretical results.

4.2 Application to Markov games

Since there are only two players, and the game is zero-sum,

$$\boldsymbol{r}^1(s, \boldsymbol{a}) = -\boldsymbol{r}^2(s, \boldsymbol{a}) \quad \forall s \in S, \boldsymbol{a} \in A,$$

therefore the payoff functions f^1 and f^2, cf. (2.3), are linked by the following relation

$$f^1(\boldsymbol{\sigma}) = -f^2(\boldsymbol{\sigma}) \quad \text{for all } \boldsymbol{\sigma} \in \Pi.$$

From now on we shall denote

$$\boldsymbol{r}(s, \boldsymbol{a}) := \boldsymbol{r}^1(s, \boldsymbol{a}) \quad \forall s \in S, \boldsymbol{a} \in A \tag{4.4}$$

and

$$f(\boldsymbol{\sigma}) := f^1(\boldsymbol{\sigma}), \quad \boldsymbol{\sigma} \in \Pi, \tag{4.5}$$

and by referring to the payoff function only have in mind the payoff function of player 1.

Remark 4.1 *Of course, here* $A = A^1 \times A^2$, *and* $\Pi = \Pi^1 \times \Pi^2$ *is a class of all bistrategies* $\boldsymbol{\sigma} = (\sigma^1, \sigma^2)$, *where* $\sigma^1 \in \Pi^1, \sigma^2 \in \Pi^2$.

For a fixed bistrategy $\boldsymbol{\sigma} = (\sigma^1, \sigma^2) \in \Pi$ the expected total reward to player 1 is

$$f(\boldsymbol{\sigma})(s) = E_{\boldsymbol{\sigma}} \left[\sum_{t=1}^{\infty} \beta^{t-1} \boldsymbol{r}(X_t, Y_t) \Big| X_1 = s \right] \tag{4.6}$$

as seen in Chapter 2.

For a fixed randomized Markov bidecision rule $\delta : S \to \mathbb{P}(A)$ we shall define

$$\begin{aligned} \boldsymbol{R}(s, \delta(s)) &:= \sum_{\boldsymbol{a} \in A_s} \boldsymbol{r}(s, \boldsymbol{a}) \left(\delta(s)\right)(\boldsymbol{a}) \text{ and} \\ p_{ss'}(\delta) &:= \sum_{\boldsymbol{a} \in A_s} q(s'|s, \boldsymbol{a}) \left(\delta(s)\right)(\boldsymbol{a}), \end{aligned} \tag{4.7}$$

and denote by $\boldsymbol{r}(\delta)$ the m-dimensional vector with (vector-valued) components $\boldsymbol{R}(s, \delta(s)), s \in S$, and by P_δ the $m \times m$-dimensional matrix whose (k, l)-th component is $p_{kl}(\delta)$.

Remark 4.2 *For a given* $\delta : S \to \mathbb{P}(A)$ *and a fixed discount factor* $0 \leq \beta < 1$ *the following holds*

$$h(\cdot, \delta, v) = \boldsymbol{r}(\delta) + \beta P_\delta v \in V.$$

Now let $\boldsymbol{\sigma} = (\sigma_1, \sigma_2, \dots) \in \Pi$ be a fixed Markov bistrategy. At stage t, the (k,l)-th component of the transition matrix $P^t_{\boldsymbol{\sigma}}$ is

$$P^t_{\boldsymbol{\sigma}}(k,l) = \left[P_{\sigma_t} P_{\sigma_{t-1}} \cdots P_{\sigma_1}\right](k,l) := \boldsymbol{P}_{\boldsymbol{\sigma}}(X_t = l | X_{t-1} = k),$$

where X_t is as in Section 2.5. If $v \in V$ is an arbitrary function, then

$$E_{\boldsymbol{\sigma}}\left[v(X_t)|X_1 = s\right] = P^{t-1}_{\boldsymbol{\sigma}} v(s) = \sum_{s' \in S} P^{t-1}_{\boldsymbol{\sigma}}(s,s') v(s') \tag{4.8}$$

for $t = 1, 2, \dots$, where $P^{t-1}_{\boldsymbol{\sigma}}(s,s')$ is the (s,s')-th component of the $(t-1)$-step transition matrix. This leads to the following conclusion

Proposition 4.3 *For $0 \le \beta < 1$ and $\boldsymbol{\sigma} = (\sigma_t)_{t=1,2,\dots}$ the following holds*

$$f(\boldsymbol{\sigma}) = \sum_{t=1}^{\infty} \beta^{t-1} P^{t-1}_{\boldsymbol{\sigma}} \boldsymbol{r}(\sigma_t), \tag{4.9}$$

if the limit exists.

Proof. Direct consequence of the definition of $f(\boldsymbol{\sigma})$ and (4.8). □

If we set

$$P^0_{\boldsymbol{\sigma}} \equiv I \text{ (identity matrix)},$$

it follows from (4.6) and (4.9) that

$$\begin{aligned} f(\boldsymbol{\sigma}) &= \sum_{t=1}^{\infty} \beta^{t-1} P^{t-1}_{\boldsymbol{\sigma}} \boldsymbol{r}(\sigma_t) \\ &= \boldsymbol{r}(\sigma_1) + \beta P_{\sigma_1} \boldsymbol{r}(\sigma_2) + \beta^2 P_{\sigma_1} P_{\sigma_2} \boldsymbol{r}(\sigma_3) + \cdots \\ &= \boldsymbol{r}(\sigma_1) + \beta P_{\sigma_1} \left(\boldsymbol{r}(\sigma_2) + \beta P_{\sigma_2} \boldsymbol{r}(\sigma_3) + \beta^2 P_{\sigma_2} P_{\sigma_3} \boldsymbol{r}(\sigma_4) + \cdots\right). \end{aligned}$$

Thus we have seen that

$$f(\boldsymbol{\sigma}) = \boldsymbol{r}(\sigma_1) + \beta P_{\sigma_1} f(\boldsymbol{\sigma}'), \tag{4.10}$$

where $\boldsymbol{\sigma}' = (\sigma_2, \sigma_3, \dots)$, or equivalently,

$$f(\boldsymbol{\sigma})(s) = \left[H_{\sigma_1} f(\boldsymbol{\sigma}')\right](s). \tag{4.11}$$

Specifically, if $\boldsymbol{\sigma}$ is a stationary bistrategy, that is,

$$\boldsymbol{\sigma} = (\delta, \delta, \dots) =: \delta^\infty,$$

the equation (4.10) becomes

$$f(\delta^\infty) = \boldsymbol{r}(\delta) + \beta P_\delta f(\delta^\infty).$$

With other words, $f(\delta^\infty)$ solves the system of equations

$$v = \boldsymbol{r}(\delta) + \beta P_\delta v, \tag{4.12}$$

where $v \in V$ and δ is a given randomized Markovian bidecision rule, $\delta : S \to \mathbb{P}(A)$. Equivalently, $f(\delta^\infty)$ is a unique solution of the system

$$v(s) = [H_\delta v](s), \quad s \in S.$$

Rewriting (4.12) yields

$$(I - \beta P_\delta)v = \boldsymbol{r}(\delta).$$

Since $\| P_\delta \| := \sup_{\|x\|=1} \| Px \| = 1$ and the spectral radius $\sigma(\cdot)$ of the discounted matrix βP_δ satisfies the following inequality

$$\sigma(\beta P_\delta) \leq \| \beta P_\delta \| = \beta < 1,$$

we may use Corollary C.4 in Puterman[21, p. 608] to establish that $(I - \beta P_\delta)^{-1}$ exists, so that from (4.9) we may obtain

$$v = (I - \beta P_\delta)^{-1} \boldsymbol{r}(\delta) = \sum_{t=1}^{\infty} \beta^{t-1} P_\delta^{t-1} \boldsymbol{r}(\delta) = f(\delta^\infty).$$

Thus we have proved the following theorem

Theorem 4.4 *Let $\delta^\infty := (\sigma^\infty, \rho^\infty) \in \Pi_S$. For a fixed β, $0 \leq \beta < 1$, the expected overall payoff to player 1, $f(\delta^\infty)$, is a unique solution of the equation*

$$v = H_\delta v \tag{4.13}$$

in the space V, where H_δ is given by (4.2). Moreover, $f(\delta^\infty)$ can be expressed as

$$f(\delta^\infty) = (I - \beta P_\delta)^{-1} \boldsymbol{r}(\delta).$$

This is a very useful characterization of expected payoffs and it immediately leads to the following result

Theorem 4.5 *For a given stationary bistrategy $\delta^\infty = (\sigma^\infty, \rho^\infty) \in \Pi_S$ and a fixed discount factor $0 \le \beta < 1$ the expected discounted payoff to player 1 is the unique solution v of the system*

$$v(s) = \sigma(s) B_s(v) \rho^T(s) \quad s \in S, \tag{4.14}$$

where $v \in V$. For a fixed $s \in S$, $B_s(v)$ is a matrix with p-dimensional vector-valued components, whose (k,l)-th element is

$$\boldsymbol{r}(s, (a_k^1, a_l^2)) + \beta \sum_{s' \in S} q(s'|s, (a_k^1, a_l^2)) v(s'), \quad a_k^1 \in A_s^1, a_l^2 \in A_s^2,$$

for all $k = 1, \ldots, K_s^1$, $l = 1, \ldots, K_s^2$.

Proof. We only have to rewrite the system (4.13) to obtain

$$\begin{aligned} v(s) &= \boldsymbol{r}(s, \delta(s)) + \beta \sum_{s' \in S} p_{ss'}(\delta) v(s') \\ &= \sum_{\boldsymbol{a} \in A_s} \left(\boldsymbol{r}(s, \boldsymbol{a}) + \beta \sum_{s' \in S} q(s'|s, \boldsymbol{a}) v(s') \right) \delta(s)(\boldsymbol{a}) \\ &= \sum_{a^1 \in A_s^1} \sum_{a^2 \in A_s^2} \left(\boldsymbol{r}(s, (a^1, a^2)) + \beta \sum_{s' \in S} q(s'|s, (a^1, a^2)) v(s') \right) \cdot \\ &\quad \cdot \left(\sigma(s)(a^1)\right) \left(\rho(s)(a^2)\right) \\ &= \sigma(s) B_s(v) \rho^T(s), \end{aligned}$$

where $B_s(v)$ is a matrix with the (k,l)-th element

$$\boldsymbol{w}_{kl} = \boldsymbol{r}(s, (a_k^1, a_l^2)) + \beta \sum_{s' \in S} q(s'|s, (a_k^1, a_l^2)) v(s'),$$

for all $k = 1, \ldots, K_s^1$, $l = 1, \ldots, K_s^2$. This proves the theorem. □

Remark 4.6 *The matrix $B_s(v)$ is an $K_s^1 \times K_s^2 \times p$ array made up of a $K_s^1 \times K_s^2$ array of p-dimensional vectors $\boldsymbol{w}_{kl} = (w_{kl}^i)_{i=1}^p$. The multiplication of this matrix with some (probability) vector is to be understood in the sense of Seber and Wild[25, Sec. B4]:*

If σ and ρ are K_s^1- and K_s^2-dimensional vectors, respectively, then the product

$$\sigma B_s(v) \rho^T$$

denotes the vector with the i-th component

$$\sum_{k=1}^{K_s^1} \sum_{l=1}^{K_s^2} \sigma_k w_{kl}^i \rho_l.$$

It is a direct consequence of Theorem 4.5 that

Corollary 4.7 *D-equilibria of a two-player zero-sum Markov game are non-dominated solutions of the system (4.14) with respect to the preference relation induced by D.*

This is a very important result, which could be interpreted as a vector-valued generalization of Shapley's result. Indeed, for scalar-valued games, that is $p = 1$, the above corollary states, that the optimal solution of the game is the (unique) solution of the system (4.14), where the space V now represents the space of all bounded scalar-valued functions. This corresponds to Theorem 1 in Shapley[22, p.203].

With the above results we can develop algorithms for solving two-person zero-sum vector-valued Markov games. We shall limit ourselves to the search of Pareto optima, that is,

Assumption. Let the preference relation be induced by the Pareto cone, $D = (0, \infty)^p$.

4.3 Successive approximations

We have already seen that the solutions of vector-valued problems may be obtained through scalarization of the original payoff functions. In this section we shall adopt the approach of White and Kim[29] and show that their results can be applied on two-person zero-sum Markov games, too.

With the same arguments as in White and Kim[29, p. 131, Lemma 1], but appropriately modified operator definitions, we can limit ourselves (in the scalarizing procedure) to the subset

$$\mathcal{A} := \{\alpha = (\alpha_1, \ldots, \alpha_p) \in \mathbb{R}^p | \sum_{k=1}^{p} \alpha_k = 1, \alpha \stackrel{\geq}{=} \mathbf{0}\} \tag{4.15}$$

of $\mathbb{R}^p$ in seeking all nondominated solutions of the system (4.13).

Let $W = \{w : S \times \mathcal{A} \to \mathbb{R} | w \text{ bounded}\}$ be the space of all bounded scalar-valued functions on $S \times \mathcal{A}$. By taking scalarizing factors into consideration we shall have to extend the definition of a decision rule as follows:

Definition 4.8 *A **(randomized Markovian) decision rule in a scalarized game** is a function*

$$\xi : S \times \mathcal{A} \to \mathbb{P}(A),$$

where $\xi(s, \alpha) \in \mathbb{P}(A_s)$ for all $s \in S$ and $\alpha \in \mathcal{A}$.

Furthermore, for a fixed bidecision rule $\xi = (\xi^1, \xi^2)$, we define a dynamic programming operator $G_\xi : W \to W$ to be

$$[G_\xi w](s, \alpha) := \langle \boldsymbol{R}(s, \xi(s, \alpha)),\, \alpha \rangle + \beta \sum_{s' \in S} p_{ss'}(\xi)\, w(s', \alpha) \tag{4.16}$$

for all $w \in W$, where $\boldsymbol{R}(s, \xi(s, \alpha))$ and $p_{ss'}(\xi)$ are defined as in (4.7) with δ replaced by $\xi(\cdot, \alpha)$. We also let $G_* : W \to W$ denote the operator, given by

$$[G_* w](s, \alpha) := \max_{\xi^1} \min_{\xi^2} [G_\xi w](s, \alpha).$$

It is easily shown that G_ξ are contractions for all ξ with respect to the supremum norm on W. That G_* is also a contraction follows from Shapley[22]. Therefore there exist fixed points g_ξ and g_* of the operators G_ξ and G_*, respectively.

The successive approximations technique is based on the fact that, since G_* is a contraction, the sequence

$$g_n = G_* g_{n-1}, \quad n \geq 1,$$

where g_0 is arbitrary, converges uniformly to g_*.

Now let for some $n \in \mathbb{N}$ g_n be linear in α, that is,

$$g_n(s, \alpha) = \langle h_n(s), \alpha \rangle \text{ for some } h_n \in V.$$

Then

$$\begin{aligned} g_{n+1}(s, \alpha) &= \max_{\xi^1} \min_{\xi^2} \left[\langle \boldsymbol{R}(s, \xi(s, \alpha)), \alpha \rangle + \beta \sum_{s' \in S} p_{ss'}(\xi) \langle h_n(s'),\, \alpha \rangle \right] \\ &= \max_{\xi^1} \min_{\xi^2} \langle \boldsymbol{R}(s, \xi(s, \alpha)) + \beta \sum_{s' \in S} p_{ss'}(\xi) h_n(s'),\, \alpha \rangle \qquad (4.17) \\ &= \max_{\xi^1} \min_{\xi^2} \langle \gamma_s(\xi),\, \alpha \rangle, \end{aligned}$$

where

$$\gamma_s(\xi) := \boldsymbol{R}(s, \xi(s, \alpha)) + \beta \sum_{s' \in S} p_{ss'}(\xi) h_n(s'). \tag{4.18}$$

Because g_{n+1} is obtained as the reward of a zero-sum game, determined by the system (4.14), and because an equilibrium decision rule in a two-person zero-sum game always exists(cf. Appendix A), there exists a ξ_{n+1}, such that

$$g_{n+1}(s, \alpha) = \langle \gamma_s(\xi_{n+1}),\, \alpha \rangle.$$

This means that g_{n+1} is also linear in α and consequently,

Proposition 4.9 *Let $g_0 \in W$ be a linear function in α, that is*

$$g_0(s, \alpha) = \langle h_0(s), \alpha \rangle \text{ for some } h_0 \in V.$$

Then each approximation of g_* is also linear in α.

Proof. Follows from (4.17) with complete induction. □

For a fixed iteration step $n \in \mathbb{N}$ and fixed $s \in S$ the optimal decision rule ξ_n, which generates g_n, defines a partition $\{\mathcal{R}^\delta_{ns}\}_\delta$ of the set $\mathcal{A}$ through

$$\mathcal{R}^\delta_{ns} := \{\alpha \in \mathcal{A} | \xi_n(s, \alpha) = \delta(s)\}, \quad \delta \in \Pi_S.$$

For fixed n and s, such collection partitions the set $\mathcal{A}$ into subsets on which $\xi_n(s, \alpha)$ is constant or, with other words, independent of α. If, for a fixed iteration step n, we collect all $\{\mathcal{R}^\delta_{ns}\}_\delta$ into one single partition, we obtain some collection $\mathcal{R}_n := \{\mathcal{R}^\delta_n\}_\delta$, and refer to it as the partition of the set $\mathcal{A}$, generated by $\{\mathcal{R}^\delta_{ns}\}_\delta$, $s \in S$.

Now we shall demonstrate how the partition of $\mathcal{A}$ on the next iteration step can be obtained from $\{\mathcal{R}^\delta_n\}_\delta$. For a fixed δ, such that

$$\delta = \xi(\cdot, \alpha), \quad \alpha \in \mathcal{R}^\delta_n,$$

the function g_n takes the form

$$g_n(s, \alpha) = \langle \gamma_s(\delta),\, \alpha \rangle$$

for all $\alpha \in \mathcal{R}_n^\delta$, where $\gamma_s(\delta)$ is as in (4.18). Thus the iterate on the next step is

$$\begin{aligned} g_{n+1}(s,\alpha) &= \max_{\xi^1}\min_{\xi^2}\left[\langle \boldsymbol{R}(s,\xi(s,\alpha)),\ \alpha\rangle + \beta\sum_{s'\in S} p_{ss'}(\xi) g_n(s',\alpha)\right] \\ &= \max_{\xi^1}\min_{\xi^2}\left[\langle \boldsymbol{R}(s,\xi(s,\alpha)),\ \alpha\rangle + \beta\sum_{s'\in S} p_{ss'}(\xi)\langle\gamma_{s'}(\delta),\ \alpha\rangle\right] \\ &= \max_{\xi^1}\min_{\xi^2}\langle \boldsymbol{R}(s,\xi(s,\alpha)) + \beta\sum_{s'\in S} p_{ss'}(\xi)\gamma_{s'}(\delta),\ \alpha\rangle \\ &= \max_{\xi^1}\min_{\xi^2}\langle \gamma^\delta_{n+1,s}(\xi),\ \alpha\rangle \\ &= \langle \gamma^\delta_{n+1,s}(\xi_{n+1}),\ \alpha\rangle, \end{aligned}$$

where

$$\gamma^\delta_{n+1,s}(\xi) := \boldsymbol{R}(s,\xi(s,\alpha)) + \beta\sum_{s'\in S} p_{ss'}(\xi)\gamma_{s'}(\delta).$$

The optimal decision rule ξ_{n+1} generates a partition of the set $\mathcal{R}_n^\delta$, denoted by $\{\mathcal{R}_{ns}^{\vartheta\delta}\}_\vartheta$, where

$$\begin{aligned} \mathcal{R}_{ns}^{\vartheta\delta} &:= \{\alpha\in\mathcal{R}_{ns}^\delta | \langle\gamma^\vartheta_{n+1,s}(\xi_{n+1}),\ \alpha\rangle = \max_{\xi^1}\min_{\xi^2}\langle\gamma^\delta_{n+1,s}(\xi),\ \alpha\rangle\} \\ &= \{\alpha\in\mathcal{R}_{ns}^\delta | \vartheta(s) = \xi_{n+1}(s,\alpha)\}, \quad \vartheta\in\Pi_S \end{aligned} \tag{4.19}$$

for fixed $n\in\mathbb{N}$ and $s\in S$. Let $\{\mathcal{R}_{n+1}^{\vartheta\delta}\}_\vartheta$ be the partition of $\mathcal{R}_n^\delta$, generated by $\{\mathcal{R}_{ns}^{\vartheta\delta}\}_\vartheta$ and denote

$$\mathcal{R}_{n+1} := \{\mathcal{R}_{n+1}^{\vartheta\delta}\}_{\vartheta,\delta}.$$

Thus the successive approximation algorithm is as follows:

1. Initialize $n=0$, $g_0(s,\alpha) = \langle h(s),\ \alpha\rangle$, and set $\mathcal{R}_0 = \{\mathcal{A}\}$, $\epsilon > 0$.
2. Given: $g_n(s,\alpha)$, and the partition $\mathcal{R}_n$.
3. Choose an element of the partition $\mathcal{R}_n$, say $\mathcal{R}_n^\delta$.
4. Choose $s\in S$.
5. Set up the matrix $\bar{B}_s(g_n,\alpha)$, and find the optimal decision rule ξ_{n+1} in the corresponding matrix game.
6. Subpartition $\mathcal{R}_n^\delta$ as in (4.19) to obtain $\{\mathcal{R}_{ns}^{\vartheta\delta}\}_\vartheta$.

7. If all the states $s \in S$ have been considered, proceed with the next step, otherwise return to step 4.

8. Find $\{\mathcal{R}_{n+1}^{\vartheta\delta}\}_{\vartheta}$ from $\{\mathcal{R}_{ns}^{\vartheta\delta}\}_{\vartheta}$, $s \in S$ and determine $g_{n+1}(s,\alpha)$.

9. If all the elements of the partition $\mathcal{R}_n$ have been considered, set $\mathcal{R}_{n+1} = \{\mathcal{R}_{n+1}^{\vartheta\delta}\}_{\vartheta,\delta}$ and proceed with the next step, otherwise return to step 3.

10. If

$$\left\|\xi_{n+1} - \xi_n\right\|_{\infty} \leq \epsilon$$

stop, otherwise set $n \to n+1$ and return to step 2.

Remark 4.10 *1. Here, $\|\cdot\|_{\infty}$ signifies the maximum norm.*

2. As a consequence of the nature of the problem, one cannot expect that, in general, the partitions obtained with the above algorithm shall be finite.

Thus the developed algorithm produces finer partitions of the set $\mathcal{A}$ on each iteration step. What remains after convergence is reached, is a partition with optimal strategies already determined in the course of the algorithm.

With respect to bounds on $\| g - g_n \|_{\mathcal{A}} := \max_{s\in S, \alpha\in\mathcal{A}} |g(s,\alpha) - g_n(s,\alpha)|$ it is shown in Puterman[21, Theorem 6.3.3, p. 163] that

$$\| g - g_n \|_{\mathcal{A}} \leq \frac{\beta^n}{1-\beta} \| g_1 - g_0 \|_{\mathcal{A}} .$$

4.4 A Hoffman-Karp-type algorithm

In Section 4.2 we have introduced the notation $B_s(v)$ for a matrix with vector-valued components

$$\boldsymbol{w}_{kl} = \boldsymbol{r}(s,(a_k^1,a_l^2)) + \beta \sum_{s'\in S} q(s'|s,(a_k^1,a_l^2))v(s'), \quad a_k^1 \in A_s^1, a_l^2 \in A_s^2, s \in S.$$

Let $g : S \times A \to \mathbb{R}$ denote some scalar-valued function on $S \times A$. Denote in this section by $\bar{B}_s(g,\alpha)$ a matrix with scalar-valued components

$$\langle \boldsymbol{r}(s,(a_k^1,a_l^2)),\alpha\rangle + \beta \sum_{s'\in S} q(s'|s,(a_k^1,a_l^2))g(s'), \quad a_k^1 \in A_s^1, a_l^2 \in A_s^2, \qquad (4.20)$$

for $s \in S$ and $\alpha \in \mathcal{A}$, where $\mathcal{A}$ is as in the previous section.

As a parallel to the policy iteration algorithm for Markov Decision Processes, the Hoffman-Karp algorithm for scalar-valued stochastic games suggests the following procedure (for details see, e.g., Breton et al.[2] or Rao, Chandrasekaran, Nair[23]):

Start with a fixed strategy ρ_0 of player 2 and solve the resulting Markov decision processfor player 1 to obtain the approximative value $\bar{g}_0$ of the optimal payoff. Now set up the system of matrix games $B_s(\bar{g}_0)$ as proposed by Shapley[22], and solve it to obtain $\delta_1 = (\sigma_1, \rho_1)$ and g_1. Repeat the procedure until convergence is reached.

As it has been shown by Rao et al.[23], the above algorithm converges, which makes it possible for us to apply the same line of reasoning to vector-valued Markov games. Again, with the same arguments as in the previous section, we can limit ourselves to the set $\mathcal{A}$ as given by (4.15), and let $\xi : S \times \mathcal{A} \to \mathbb{P}(A)$ denote a randomized Markovian bidecision rule in the scalarized game.

Let for a fixed iteration step $n \in \mathbb{N}$ the bidecision rule ξ_n and some partition $\mathcal{R}_n$ of the set $\mathcal{A}$ be given. On a chosen element of $\mathcal{R}_n$, say $\mathcal{R}_n^{\underline{\delta}}$, the following holds:

$$\begin{aligned} \xi_n(s,\alpha) = \underline{\delta}(s) &= (\underline{\sigma}(s), \underline{\rho}(s)) \\ v_n(s) &= \underline{\sigma}(s) B_s(v_{n-1}) \underline{\rho}^T(s), \quad s \in S. \end{aligned}$$

Now keep $\underline{\rho}$ fixed, and solve the vector-valued MDP

$$v(s) = \max_{\sigma \in \Pi^1} \sigma(s) B_s(v) \underline{\rho}^T(s), \quad s \in S,$$

(here, 'max' is meant in the sense of vector optimization) for player 1, for example with one of the algorithms supplied by White and Kim[29], to obtain $\bar{v}_n(s,\alpha)$ and the corresponding partition $\{\mathcal{R}_n^{\underline{\delta},a}\}_a$ of the set $\mathcal{R}_n^{\underline{\delta}}$. At this step $\bar{v}_n(\cdot,\alpha)$ is constant on each $\mathcal{R}_n^{\underline{\delta},a}$. Set up the system of vector-valued matrix games $B_s(\bar{v}_n)$, $s \in S$, on each $\mathcal{R}_n^{\underline{\delta},a}$, and perform one step of the successive approximation algorithm to obtain the final partition $\{\mathcal{R}_n^{\underline{\delta},a,\vartheta}\}_\vartheta$ of the set $\mathcal{R}_n^{\underline{\delta},a}$. Collect all $\{\mathcal{R}_n^{\underline{\delta},a,\vartheta}\}_{\underline{\delta}\vartheta}$ into one single partition, and denote it by $\mathcal{R}_{n+1}$. Use the optimal decision rule in the matrix game $B_s(\bar{v}_n)$ as ξ_{n+1} on the next iteration step.

Thus the vector-valued Hoffman-Karp algorithm is as follows:

1. Set $n = 0, \epsilon > 0$ and $\mathcal{R} = \{\mathcal{A}\}$, and choose ρ_0.

2. Given: ρ_n and the partition $\mathcal{R}_n$.

3. Choose an element of the partition $\mathcal{R}_n$, say $\mathcal{R}_n^{\delta}$.

4. Solve the Markov decision problem for player 1 to obtain $\sigma_n, \bar{v}_n$, and the partition of the set $\mathcal{R}_n^{\delta}$. Denote this partition by $\mathcal{R}_n^{1,k} = \{\mathcal{R}_n^{\delta,a}\}_{a \in A}$.

5. Choose $s \in S$.

6. Choose an element of the partition $\mathcal{R}_n^{1,k}$, say $\mathcal{R}_n^{\delta,a}$. Set up the matrix game $B_s(\bar{v}_n)$ and use one step of the successive approximations algorithm to obtain the optimal reward $v_n(s,\alpha)$, the optimal strategy $\xi(s,\alpha)$, and a new partition of the set $\mathcal{R}_n^{\delta,a}$, and denote it by $\mathcal{R}_{n,s}^{\delta,a}$.

7. If all the elements of the partition $\mathcal{R}_n^{1,k}$ have been considered, proceed with the next step, otherwise return to step 6.

8. If all elements of the partition $\mathcal{R}_n$ have been considered, collect all newly formed partitions into $\mathcal{R}_{n+1}$ and proceed with the next step, otherwise return to step 3.

9. If

$$\left\| \xi_{n+1} - \xi_n \right\|_{\infty} \leq \epsilon$$

stop, otherwise set $n \to n+1$, $\xi_{n+1}(s,\alpha) = \xi(s,\alpha)$, and return to step 2.

4.5 An example

In this section we present a numerical example for both the successive approximations and the Hoffman-Karp algorithm. Because of the nature of the problem, i.e. the difficulties that arise at the algorithm implementation, and thus their computational inefficiency, we shall only present the first step of both. The problem is defined as follows:

Let there be two states in the two-person zero-sum game, i.e.,

$$S = \{1, 2\}, \quad m = 2,$$

two actions for each player in each state,

$$A_1^1 = \{a_1, b_1\} \qquad A_2^1 = \{c_1, d_1\}$$
$$A_1^2 = \{a_2, b_2\} \qquad A_2^2 = \{c_2, d_2\},$$

and therefore (cf. Section 2.2),

$$A_1 = \{(a_1, a_2), (a_1, b_2), (b_1, a_2), (b_1, b_2)\}$$
$$A_2 = \{(c_1, c_2), (c_1, d_2), (d_1, c_2), (d_1, d_2)\}.$$

Let the transition probabilites be

$$\boldsymbol{q}(1, (a_1, a_2)) = (\frac{1}{3}, \frac{2}{3}) \qquad \boldsymbol{q}(1, (a_1, b_2)) = (\frac{2}{3}, \frac{1}{3})$$
$$\boldsymbol{q}(1, (b_1, a_2)) = (\frac{1}{2}, \frac{1}{2}) \qquad \boldsymbol{q}(1, (b_1, b_2)) = (\frac{1}{4}, \frac{3}{4})$$
$$\boldsymbol{q}(2, (c_1, c_2)) = (\frac{1}{3}, \frac{2}{3}) \qquad \boldsymbol{q}(2, (c_1, d_2)) = (\frac{2}{3}, \frac{1}{3})$$
$$\boldsymbol{q}(2, (d_1, c_2)) = (\frac{1}{2}, \frac{1}{2}) \qquad \boldsymbol{q}(2, (d_1, d_2)) = (\frac{1}{4}, \frac{3}{4}),$$

and the payoff matrices take the form

$$s = 1: \qquad \begin{pmatrix} (4, 8, 12) & (3, 7, 16) \\ (8, 3, 10) & (5, 12, 3) \end{pmatrix} \qquad s = 2: \qquad \begin{pmatrix} (6, 8, 10) & (12, 10, 5) \\ (9, 15, 2) & (8, 9, 6) \end{pmatrix},$$

where the payoffs in the state $s = 1$ are aligned as follows:

$$\begin{pmatrix} \boldsymbol{r}(1, (a_1, a_2)) & \boldsymbol{r}(1, (a_1, b_2)) \\ \boldsymbol{r}(1, (b_1, a_2)) & \boldsymbol{r}(1, (b_1, b_2)) \end{pmatrix}$$

and similarly for the payoff matrix in the state $s = 2$. Finally, let $\beta = 0.8$.

The resulting partitions of the set $\mathcal{A}$ shall be presented in the barycentric coordinates.

4.5.1 Successive approximations

For the initial iteration value in the successive approximations algorithm we choose

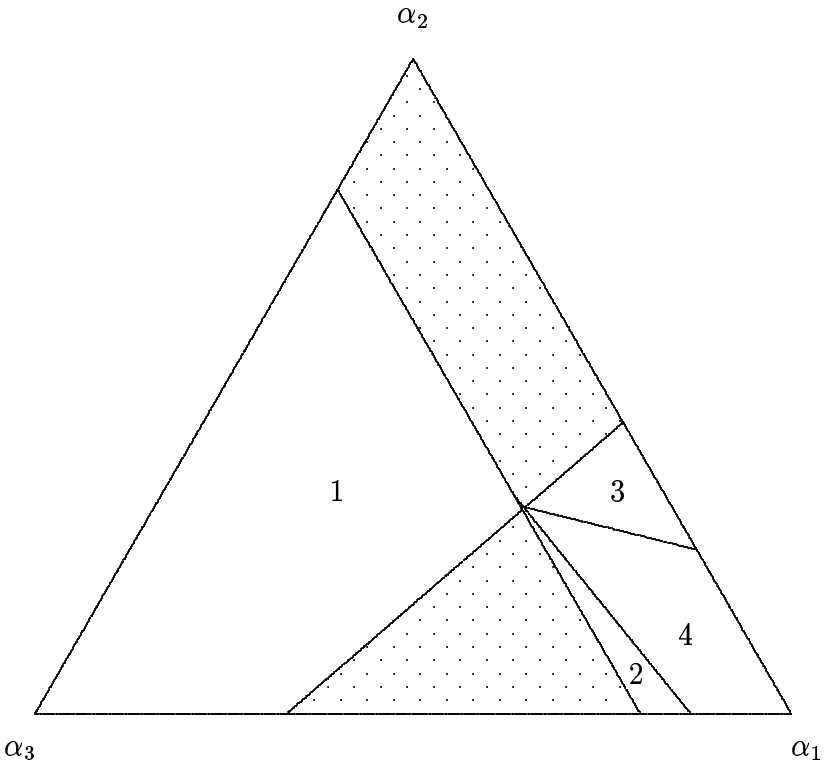
$$g_0(s, \alpha) \equiv 0 \text{ and } \mathcal{R}_0 = \{\mathcal{A}\}.$$

Thus, at iteration step $n = 1$, one has to solve two matrix games for a fixed α. Because clearly both states represent 2×2 matrix games, namely

$$\bar{B}_1(g_0, \alpha): \qquad \begin{pmatrix} \langle (4, 8, 12), \alpha \rangle & \langle (3, 7, 16), \alpha \rangle \\ \langle (8, 3, 10), \alpha \rangle & \langle (5, 12, 3), \alpha \rangle \end{pmatrix} \qquad \bar{B}_2(g_0, \alpha): \qquad \begin{pmatrix} \langle (6, 8, 10), \alpha \rangle & \langle (12, 10, 5), \alpha \rangle \\ \langle (9, 15, 2), \alpha \rangle & \langle (8, 9, 6), \alpha \rangle \end{pmatrix},$$

one can use explicit formulas to obtain pure and mixed strategies (in the sense of static Game Theory).

With these results, the partition $\{\mathcal{R}_{01}^{\vartheta}\}_{\vartheta}$ for $\vartheta = \xi(\cdot, \alpha)$, obtained in the state $s = 1$, is



where the mixed strategies are given by

$$\begin{aligned} \xi^1(1,\alpha) &= \frac{1}{\langle(-2,10,-11),\,\alpha\rangle}\Big(\langle(-3,9,-7),\,\alpha\rangle, \langle(1,1,-4),\,\alpha\rangle\Big) \\ \xi^2(1,\alpha) &= \frac{1}{\langle(-2,10,-11),\,\alpha\rangle}\Big(\langle(2,5,-13),\,\alpha\rangle, \langle(-4,5,2),\,\alpha\rangle\Big) \end{aligned} \tag{4.21}$$

for players 1 and 2, respectively, and appear shaded in the above figure, and the numbers represent the strategies, presented in Table 4.1.

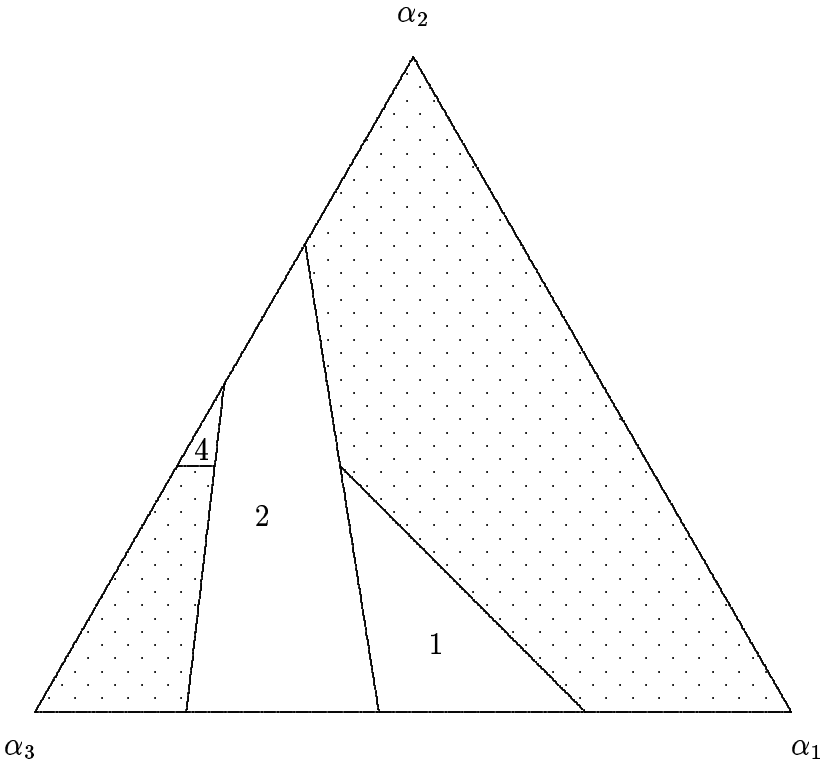
Similarly we obtain the formulae for mixed strategies in state $s = 2$:

$$\begin{aligned} \xi^1(2,\alpha) &= \frac{1}{\langle(-7,-8,9),\,\alpha\rangle}\Big(\langle(-1,-6,4),\,\alpha\rangle, \langle(-6,-2,5),\,\alpha\rangle\Big) \\ \xi^2(2,\alpha) &= \frac{1}{\langle(-7,-8,9),\,\alpha\rangle}\Big(\langle(-4,-1,1),\,\alpha\rangle, \langle(-3,-7,8),\,\alpha\rangle\Big) \end{aligned} \tag{4.22}$$

Table 4.1: Optimal strategies

Number	Strategy $\xi(1,\alpha)$	Strategy $\xi(2,\alpha)$
1	((1,0),(1,0))	((1,0),(1,0))
2	((1,0),(0,1))	((1,0),(0,1))
3	((0,1),(1,0))	((0,1),(1,0))
4	((0,1),(0,1))	((0,1),(0,1))

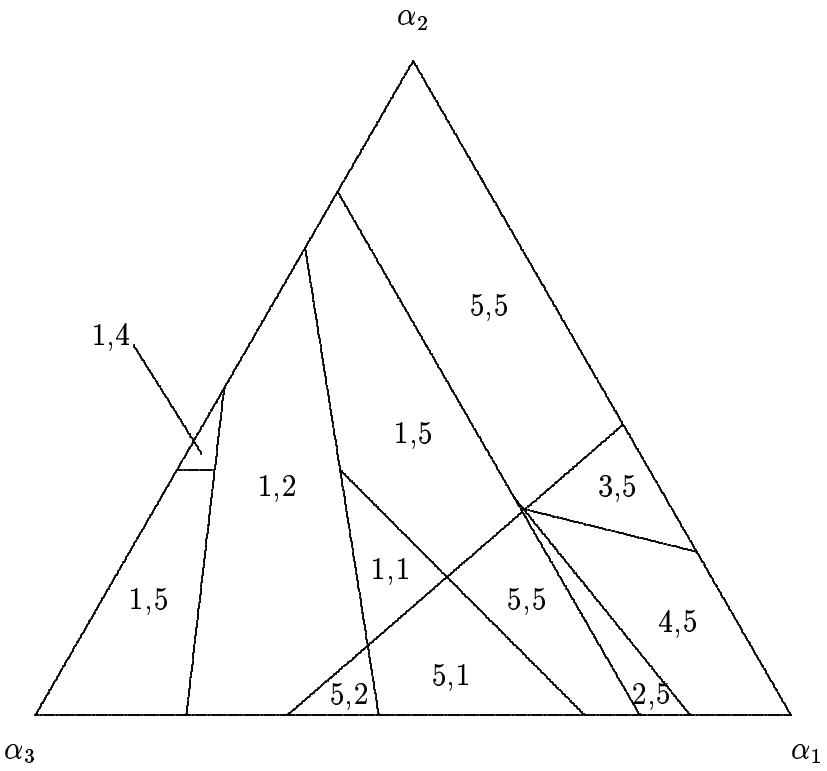
for players 1 and 2, respectively, and the partition thus induced, namely $\{\mathcal{R}_{02}^{\vartheta}\}_{\vartheta}$ for $\vartheta = \xi(\cdot,\alpha)$, is



where the legend is as in Table 4.1. As in Step 8 of the algorithm we now combine both partitions to finally obtain $\mathcal{R}_1$ as represented in the next figure. Here, the numbers represent the strategy pairs

$$\xi(\alpha) = (\xi(1,\alpha), \xi(2,\alpha))$$

with number 5 representing the mixed strategies as given by (4.21) and (4.22), and the rest of the legend is as in Table 4.1.



It is clear what the payoff to player 1 under the optimal strategies 1 to 4 in either state is. The payoff for the elements of $\mathcal{R}_1$, labelled by 5, is given by

$$g_1(i,\alpha) = \xi^1(i,\alpha)\bar{B}_i(g_0,\alpha)\xi^2(i,\alpha)^T, \quad i = 1,2.$$

Clearly, these values strongly depend on the choice of α. The reason for this is, that here, ξ is not independent of α, because we obtain a different value of ξ for each value of α. Therefore the representation in the above figure is only schematic. One should be aware of the fact, that the elements of $\mathcal{R}_1$, labelled by 5 are composed of infinitely many sets, on which ξ is independent of α. These elements may also be singletons.

This also makes our algorithms in the present form not particularly useful for direct computation. The formulae for optimal strategies in the above example, however, are fairly easy to determine, as we are in fact solving 2×2 matrix games, where explicit formulae for mixed strategies are well-known. Thus we can, in this simple example, obtain analytical solutions to the game.

In general, one can now proceed with the iteration to obtain finer partitions and better approximations of g_* at each iteration step.

4.5.2 Hoffman-Karp algorithm

Here, at iteration step $n = 0$, we choose

$$\rho_0(1) = (1, 0), \quad \rho_0(2) = (1, 0),$$

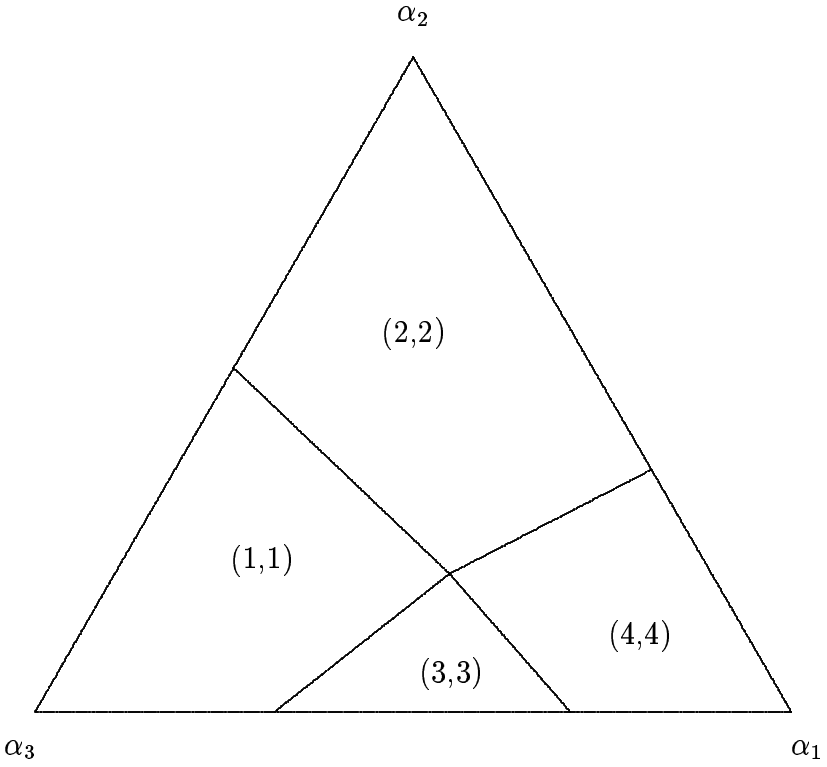
meaning that the chosen strategies of player 2 at this step are pure (deterministic) strategies in each state $s = 1, 2$.

Player 1 now attempts to find nondominated solutions of the following system

$$v(1) = \sigma(1) \begin{pmatrix} (4, 8, 12) + 0.8(\frac{1}{3}v(1) + \frac{2}{3}v(2)) \\ (8, 3, 10) + 0.8(\frac{1}{2}v(1) + \frac{1}{2}v(2)) \end{pmatrix}$$
$$v(2) = \sigma(2) \begin{pmatrix} (6, 8, 10) + 0.8(\frac{1}{3}v(1) + \frac{2}{3}v(2)) \\ (9, 15, 2) + 0.8(\frac{1}{2}v(2) + \frac{1}{2}v(2)) \end{pmatrix},$$

for $v \in V$, as determined by means of formula (4.14).

To solve this vector-valued MDP, we have used the 'direct search algorithm,' proposed by White and Kim[29], and obtained the following partition $\mathcal{R}_0^{1,1}$ of the set $\mathcal{A}$:



The optimal strategies, and their notations, for player 1 are presented in Table 4.2.

Table 4.2: Optimal strategies for the MDP

Notation	Strategy $\xi^1(1,\alpha)$	Strategy $\xi^1(2,\alpha)$
(1,1)	(1,0)	(1,0)
(2,2)	(1,0)	(0,1)
(3,3)	(0,1)	(1,0)
(4,4)	(0,1)	(0,1)

After the MDP has been solved, we proceed with one step of the successive approximation algorithm on every element of the partition $\mathcal{R}_0^{1,1}$.

Let us consider the element of $\mathcal{R}_0^{1,1}$, labelled by $(2,2)$. Here, the optimal payoff $\bar{v}$ is given by

$$\bar{v}(1) = (31.7647, 56.4706, 36.4706)$$
$$\bar{v}(2) = (36.1765, 62.6471, 27.6471).$$

Therefore we now attempt to find nondominated solutions of the vector-valued matrix games

$$B_1(\bar{v}) = \begin{pmatrix} (31.7647, 56.4706, 36.4706) & (29.5882, 53.8235, 42.8235) \\ (35.1765, 50.6471, 35.6471) & (33.0588, 60.8824, 26.8824) \end{pmatrix}$$

and

$$B_2(\bar{v}) = \begin{pmatrix} (33.7647, 56.4706, 34.4706) & (38.5882, 56.8235, 31.8235) \\ (36.1765, 62.6471, 27.6471) & (36.0588, 57.8824, 29.8824) \end{pmatrix}.$$

After performing one step of the successive approximations algorithm on $(2,2)$ we obtain the partition

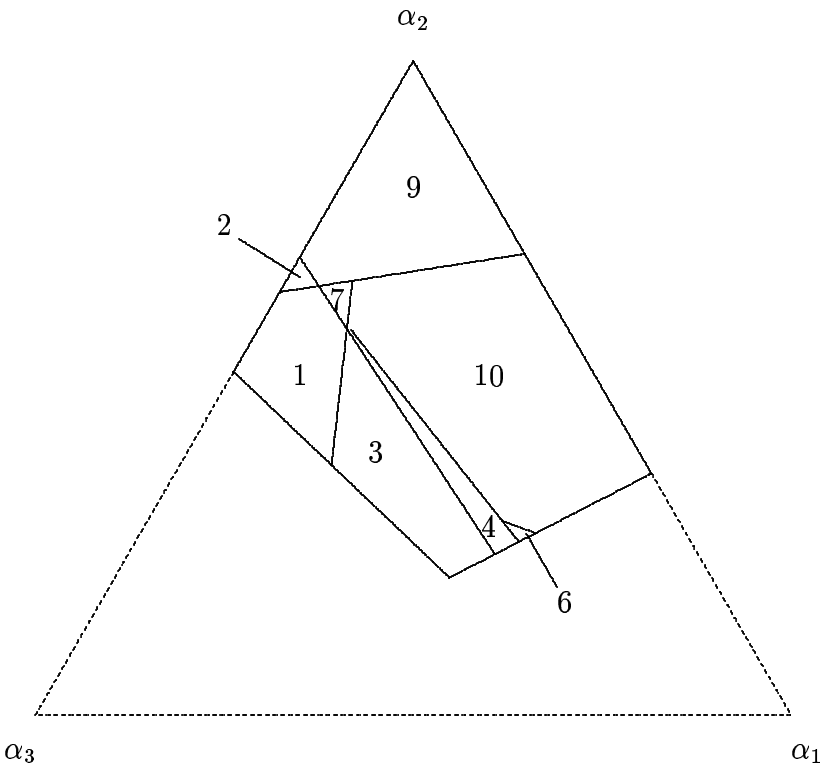


Table 4.3: Optimal strategies for the H-K algorithm

Label	$\xi(1,\alpha)$	$\xi(2,\alpha)$
1	((1,0),(1,0))	((1,0),(0,1))
2	((1,0),(1,0))	((0,1),(0,1))
3	((1,0),(1,0))	mixed
4	((1,0),(0,1))	mixed
5	((0,1),(1,0))	mixed
6	((0,1),(0,1))	mixed
7	mixed	((1,0),(0,1))
8	mixed	((0,1),(1,0))
9	mixed	((0,1),(0,1))
10	mixed	mixed

Here, the legend is presented in Table 4.3, and the mixed strategies are given

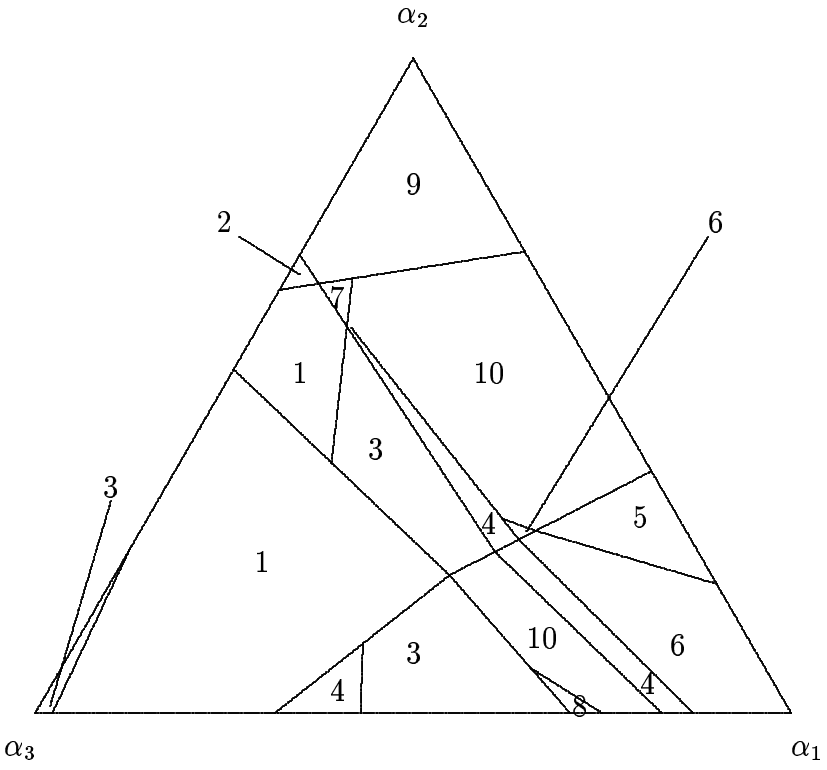
by the following formulae:

$$\xi^1(1,\alpha) = (\frac{\langle(-2.118, 10.235, -8.765), \alpha\rangle}{\langle(0.059, 12.882, -15.118), \alpha\rangle}, \frac{\langle(2.176, 2.647, -6.353), \alpha\rangle}{\langle(0.059, 12.882, -15.118), \alpha\rangle})$$
$$\xi^2(1,\alpha) = (\frac{\langle(3.471, 7.059, -15.941), \alpha\rangle}{\langle(0.059, 12.882, -15.118), \alpha\rangle}, \frac{\langle(-3.412, 5.824, 0.824), \alpha\rangle}{\langle(0.0589, 12.882, -15.118), \alpha\rangle})$$

in the state $s = 1$, and

$$\xi^1(2,\alpha) = (\frac{\langle(-0.117, -4.7647, 2.2353), \alpha\rangle}{\langle(-4.91, -5.118, 4.882), \alpha\rangle}, \frac{\langle(-4.8235, -0.3529, 2.6471), \alpha\rangle}{\langle(-4.91, -5.118, 4.882), \alpha\rangle})$$
$$\xi^2(2,\alpha) = (\frac{\langle(-2.5294, 1.0589, -1.9411), \alpha\rangle}{\langle(-4.91, -5.118, 4.882), \alpha\rangle}, \frac{\langle(-2.4118, -6.1765, 6.8235), \alpha\rangle}{\langle(-4.91, -5.118, 4.882), \alpha\rangle})$$

for $s = 2$. Repeating the procedure on all elements of $\mathcal{R}_0^{1,1}$ gives us the sought partition $\mathcal{R}_1$, presented in the following figure:



Here, the legend is as in Table 4.3.

4.6 Other results

Unfortunately, as an example in Breton et al.[2, Sec. 4.1] shows, the dynamic programming approach is not extendable to non-zero-sum stochastic games. The reason behind this is simple: already in 2×2 bimatrix games there may exist more than one equilibrium point. Therefore the dynamic programming operator in this case would neither be uniquely determined, nor monotone.

One other possible approach to solving non-zero-sum games would then be nonlinear programming. Rustem[24] proposed various algorithms to obtain equilibria of static games. Since our game (2.1) differs from static games only in model assumptions, and the basic structure remains the same (cf. equilibria definitions in Chapter 3), we can apply his results to our game model with some additional assumptions.

Firstly, we assume the payoff functions $f^1, \ldots, f^N$ to be twice continuously differentiable on Π. For general N-person games we defined an equilibrium $\boldsymbol{\sigma}_*$ to be the point, from which no player would wish to deviate unilaterally, since such an action would not improve his position, regarding his payoff. If we again restrict to the search of Pareto-optimal solutions, an equilibrium $\boldsymbol{\sigma}_* = (\sigma_*^1, \ldots, \sigma_*^N)$ is defined by

$$f^i(\sigma_*^1, \ldots, \sigma_*^{i-1}, \sigma^i, \sigma_*^{i+1}, \ldots, \sigma_*^N) \overset{\leq}{=} f^i(\boldsymbol{\sigma}_*)$$

for all $\sigma^i \in \Pi^i$ and all $i = 1, \ldots, N$.

Hence, $\boldsymbol{\sigma}_*$ is an unconstrained equilibrium of f, with respect to σ, simultaneously for all $i = 1, \ldots, N$. Because f is twice continuously differentiable, Theorem 3.23 modifies into

$$\boldsymbol{\sigma}_* \text{ equilibrium} \iff \nabla_{\sigma^i} f^i(\sigma^1, \ldots, \sigma^N) = 0, \quad i = 1, \ldots, N.$$

This also corresponds to the conditions of Proposition 1.2.3 in Rustem[24, p. 7]. In this case, Rustem[24, Ch. 9] proposes the following algorithm:

1. Given: $\sigma_0, \rho_0, \epsilon > 0$; set $n = 0$.
2. Compute $\sigma_{n+1} = \arg\max_{\sigma \in \Pi^1}\{f^1(\sigma, \rho_n)\}$ and $\rho_{n+1} = \arg\max_{\rho \in \Pi^2}\{f^2(\sigma_n, \rho)\}$.
3. Equilibrium check: if

$$\left\|\sigma_{n+1} - \sigma_n\right\|_\infty \leq \epsilon$$

and

$$\left\|\rho_{n+1} - \rho_n\right\|_\infty \leq \epsilon$$

stop, otherwise set $n \to n+1$ and return to step 2.

Remark 4.11 *Here,* $\|\cdot\|_\infty$ *signifies the maximum norm on* Π.

4.7 Conclusion

We have shown that dynamic programming approach of White and Kim[29] can be extended to two-person zero-sum Markov games. The developed algorithms are, however, computationally inefficient and are presented only to show that such an extension is possible. The previous section shortly deals with the nonlinear multicriteria programming approach, but the path, suggested by Theorem 3.23 remains unexplored in all generality.

The directions for further research thus include finding effective computational implementation of algorithms, presented in this chapter, and developing nonlinear programming algorithms for more general forms of vector-valued Markov games - either two-person general-sum, or N-person games.

Appendix A

General notions of Game Theory

A.1 Definition of a game

Game Theory analyses strategic decision situations. The basic ideas originate from the context of parlour games, like chess or bridge, which could generally be described in the following way:

The result of the decisions depends upon several *decision makers* (also referred to as *players*). No single player can determine the result independently from the other players. Each player is aware of this 'interdependence', and assumes, that all other players are also aware of it.

These (static) games thus consist of a sequence of personal moves made by the players, and as a result, the players are awarded with some payoff in the form of money, prestige, etc.

To provide a rigorous mathematical definition of a game, we first define a **topological (game) tree** to be a finite collection of **nodes** (also referred to as **vertices**), connected by lines, called **arcs**. We also demand, that for any vertices A, B there be a *unique* sequence of arcs and nodes joining A and B. As a consequence, this means that a game tree is a connected figure which includes no simple connected arcs.

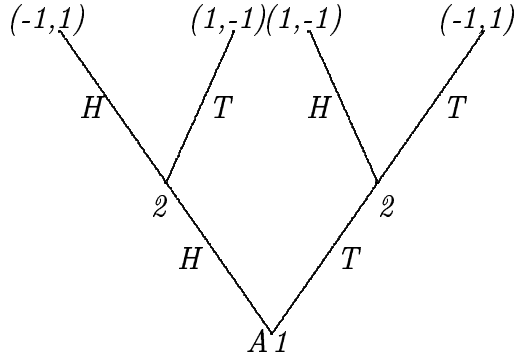
Definition A.1 *Let Γ be a topological tree with a distinguished vertex A. A vertex C is said to* ***follow*** *the vertex B if the sequence of arcs joining A to C passes through B. C follows B* ***immediately*** *if C follows B and there is*

an arc joining B and C. A vertex X is ***terminal*** *if no vertex follows X.*

Definition A.2 *An N-**person game in extensive form** is a collection of*

- *a topological tree Γ with a distinguished vertex A, called the **starting point** of Γ*
- *a function, called the **payoff function**, which assigns an N-vector to each terminal vertex of Γ*
- *a partition of the nonterminal vertices of Γ into $N+1$ sets $S_0, S_1, \ldots, S_N$, called the **player sets***
- *a probability distribution, defined at each vertex of S_0, among the immediate followers of this vertex*
- *for each $i = 1, \ldots, N$, a subpartition of S_i into subsets S_i^j, called **information sets**, such that two vertices in the same information set have the same number of immediate followers and no vertex can follow another vertex in the same information set*
- *for each information set S_i^j, an index set I_i^j, together with a one-to-one mapping of the set I_i^j onto the set of immediate followers of each vertex of S_i^j.*

Example A.3 (Matching pennies) *Player 1 chooses heads (H) or tails (T). Player 2, not knowing player 1's choice, also chooses H or T. If the two choose alike, then player 2 wins a penny from player 1, otherwise player 1 wins a penny from player 2. The game tree is presented in the following figure:*



The vectors at the terminal vertices represent the payoff function. The numbers near the other vertices denote the player to whom the move corresponds.

The purpose of playing games for each player is maximizing their payoff. To achieve this goal, each player uses some **decision rule** (in static Game Theory more commonly referred to as **strategy**), which represents a plan of playing a game. Formally,

Definition A.4 *A **(deterministic, pure) strategy** of player $i = 1, \ldots,$ N is a function which assigns, to each player i's information sets S_i^j, one of the arcs which follow a representative vertex of S_i^j. We shall denote the set of all decision rules of player i by Σ_i.*

To decide, which strategy is best (i.e., yields the optimal payoff), we take the mathematical expectation of the payoff function, given that the players are using a given N-tuple $\sigma = (\sigma_1, \ldots, \sigma_N)$ of strategies, that is

$$f(\sigma) = (f_1(\sigma), \ldots, f_N(\sigma)).$$

From this, it becomes possible to tabulate the function $f(\sigma)$ for all possible values of $\sigma_1, \ldots .\sigma_N$, either in the form of a relation, or by setting up an N-dimensional array of N-vectors. This N-dimensional array is called the **Nash normal form** of the game Γ.

Example A.5 (Matching pennies revisited) *In the game of matching pennies each players has two (deterministic) strategies, H or T. The Nash normal form of this game is the matrix*

$$\begin{array}{c} \\ H \\ T \end{array} \begin{array}{c} \begin{array}{cc} H & \quad T \end{array} \\ \begin{pmatrix} (-1,1) & (1,-1) \\ (1,-1) & (-1,1) \end{pmatrix} \end{array}$$

Here each row represents a strategy of player 1, and each column represents a strategy of player 2.

Definition A.6 *Given a game Γ, a strategy N-tuple $(\sigma_1^*, \ldots, \sigma_N^*)$ is an **equilibrium**, if and only if, for any $i = 1, \ldots, N$ and $\hat{\sigma}_i \in \Sigma_i$,*

$$f_i(\sigma_1^*, \ldots, \sigma_{i-1}^*, \hat{\sigma}_i, \sigma_{i+1}^*, \ldots, \sigma_N^*) \leq f_i(\sigma_1^*, \ldots, \sigma_N^*).$$

Unfortunately, not every game has equilibrium N-tuples.

A.2 Two-person zero-sum games

Definition A.7 *A game Γ is said to be zero-sum, if and only if, at each terminal vertex, the payoff function $(p_1, \ldots, p_N)$ satisfies*

$$\sum_{i=1}^{N} p_i = 0. \tag{A.1}$$

The normal form of a finite two-person zero-sum game reduces to a matrix, A, with as many rows as player 1 has strategies and as many columns as player 2 has strategies. The expected payoff, assuming 1 chooses his i-th strategy and 2 chooses his j-th strategy, is the element a_{ij} in the i-th row and j-th column of the matrix.

A strategy pair will be an equilibrium if, and only if, the element a_{ij} corresponds to it is both the largest in its column and the smallest in its row. Such an element, if it exists, is called a **saddle point**. If the players play a game without saddle points the strategy should be chosen at random but the randomization scheme should be chosen rationally.

Definition A.8 *A **mixed strategy** for a player is a probability distribution on the set of his pure strategies.*

Let X denote the set of all mixed strategies of player 1, and Y the set of all mixed strategies of player 2. If the players 1 and 2 choose the mixed strategies $x = (x_1, \ldots, x_m)$ and $y = (y_1, \ldots, y_n)$, respectively, then the **expected payoff** will be

$$f(x, y) := \sum_{i=1}^{m} \sum_{j=1}^{n} x_i a_{ij} y_j$$

or, in matrix notation,

$$f(x, y) = xAy^T.$$

Let us denote by $A_{i.}$ the i-th row of matrix A and by $A_{.j}$ the j-th column of A. Then the following holds:

Theorem A.9 (The Minimax Theorem) *Let X denote the set of all mixed strategies of player 1, and Y the set of all mixed strategies of player 2. Then*

$$\max_{x \in X} \min_{j} xA_{.j} = min_{y \in Y} \max_{i} A_{i.}y^T. \tag{A.2}$$

Proof. See Owen[16, p. 16].

This theorem proves that every two-person zero-sum game has optimal strategies. If A is a 2×2 matrix game, the following theorem provides the laws for computing optimal strategies:

Theorem A.10 *Let A be a 2×2 matrix game. Then if A does not have a saddle point, its unique optimal strategies and value will be given by*

$$x = \frac{JA^*}{JA^*J^T},$$
$$y = \frac{A^*J^T}{JA^*J^T},$$
$$v = \frac{|A|}{JA^*J^T},$$

where A^* is the adjoint of A, $|A|$ the determinant of A, and J the vector $(1, 1)$.

Proof. See Owen[16, Sec. II.5.5].

A.3 Two-person general-sum games

In general, a finite two-person general-sum game can be expressed as a pair of $m \times n$ matrices $A = (a_{ij})$ and $B = (b_{ij})$, or equivalently as an $m \times n$ matrix (A, B) each of whose entries is an ordered pair (a_{ij}, b_{ij}). The entries a_{ij} and b_{ij} are the payoffs to the players 1 and 2, respectively, assuming they choose, respectively, their i-th and j-th pure strategies. A game in this form is called a **bimatrix game**.

Definition A.11 *A pair of mixed strategies (x^*, y^*) is an **equilibrium** if, for any other mixed strategies x and y,*

$$xAy^{*T} \leq x^*Ay^{*T}$$
$$x^*By^T \leq x^*By^{*T}.$$

Theorem A.12 *Every bimatrix game has at least one equilibrium point.*

Proof. See Owen[16, p.162].

Example A.13 (The Prisoners' Dilemma) *Consider the game*

$$\begin{pmatrix} (5,5) & (0,10) \\ (10,0) & (1,1) \end{pmatrix}.$$

In this game, it is easy to see that the second row dominates the first row, while the second column dominates the first column. Hence, the only equilibrium pair is given by the second pure strategy of each player. This gives the payoff vector $(1,1)$*. But, if both players choose the first pure strategy, the result* $(5,5)$ *is much better for both. The trouble is that each can gain by double-crossing the other.*

A.4 N-person noncooperative games

In general, there is no great difference between the theory of noncooperative games and noncooperative two-person non-zero-sum games. The principal question of noncooperative games is the existence of equilibrium N-tuples.

Theorem A.14 *Any finite N-person noncooperative game has at least one equilibrium N-tuple of mixed strategies.*

Proof. See Owen[16, Sec. X.1]

Bibliography

[1] J.-P. Aubin: *Optima and Equilibria*, An Introduction to Nonlinear Analysis. Second Edition, Springer-Verlag Berlin, 1998.

[2] M. Breton, J. Filar, A. Haurie, T. Schultz: On the Computation of Equilibria in Discounted Stochastic Dynamic Games. In: *Dynamic Games and Applications in Economics*, London, 1985, 64-87.

[3] H.W. Corley: Games With Vector Payoffs. *J. Optim. Theory Appl.*, Vol. **47**, No. **4** (1985), 491-498.

[4] R.E.Edwards: *Functional Analysis, Theory and Applications*, Dover New York, 1994.

[5] K. Fan: Minimax Theorems. *Proc. Nat. Acad. Sci. U.S.A.* **39** (1953), 42-47.

[6] Fernandez, Puerto: Vector Linear Programming in Zero-Sum Multicriteria Matrix Games. *J. Optim. Theory Appl.*, Vol. **89** (1996), 115-127.

[7] J. Filar, T. Schultz, F. Thuijsman, O.J. Vrieze: Nonlinear Programming and Stationary Equilibria in Stochastic Games. *Mathematical Programming* **50** (1991), 227-237.

[8] D. Ghose, U.R. Prasad: Solution Concepts in Two-Person Multicriteria Games. *J. Optim. Theory Appl.*, Vol. **63**, No. **2** (1998), 167-189.

[9] K. Hinderer: *Foundations of Non-stationary Dynamic Programming with Discrete Time Parameter.* Springer-Verlag Berlin, 1970.

[10] M.J. Holler, G. Illing: *Einführung in die Spieltheorie*, Springer-Verlag Berlin, 1991.

[11] J. McMillan: *Game Theory in International Economics.* Harwood Academic Publishers, Chur, 1994.

[12] J.-F. Mertens, A.Neyman: Stochastic Games. *Int. J. Game Theory* **10** (1981), 53-66.

[13] J. F. Nash: Equilibrium Points in N-Person Games. A. Rubinstein (ed.):*Game Theory in Economics*, The International Library of Critical Writings in Economics, Elgar reference collection, 1990.

[14] J. F. Nash: Non-cooperative games. A. Rubinstein (ed.): *Game Theory in Economics*, The International Library of Critical Writings in Economics, Elgar reference collection, 1990.

[15] A.S. Nowak: Existence of Equilibrium Stationary Strategies in Discounted Non-Cooperative Stochastic Games with Uncountable State Space, *J. Optim. Theory Appl.*, **45** (1985), 591-602.

[16] G. Owen: *Game Theory*, 2nd edition, Academic Press, New York, 1982.

[17] T. Parthasarathy, M. Stern: Markov Games - A Survey. in: Roxin, P. Liu and R. Sternberg (eds.): *Differential Games and Control Theory II,* New York, 1977, pp. 1-46.

[18] A. Pascoletti, P. Serafini: Scalarizing Vector Optimization Problems. *J. Optim. Theory Appl.*, Vol. **42**, No. **4** (1984), 499-524.

[19] L.A. Petrosjan, V.V. Mazalov (eds.): *Game Theory and Applications II*, Nova Science Publishers Inc., New York, 1996.

[20] M. Piškurić: On Vector-Valued MArkov Games, *Game Theory and Applications VI*, Nova Science Publishers Inc., New York, 2001.

[21] M.L. Puterman: *Markov Decision Processes; Discrete Stochastic Dynamic Programming.* Wiley Series in Probability and Mathematical Statistics, John Wiley and Sons, New York, 1994.

[22] T.E.S. Raghavan, T.S. Ferguson, T. Parthasarathy, O.J. Vrieze (eds.): *Stochastic Games and Related Topics*, Kluwer Academic Publishers, 1991.

[23] S.S. Rao, R. Chandrasekaran, K.P.K. Nair: Algorithms for Discounted Stochastic Games. *J. Optim. Theory Appl.*, Vol. **11** (1973), 627-637.

[24] B. Rustem: *Algorithms For Nonlinear Programming and Multiple Objective Decisions.* John Wiley and Sons, Chicester, 1998.

[25] G.A.F. Seber, C.J. Wild: *Nonlinear Regression*. John Wiley and Sons, New York, 1989.

[26] M.J. Sobel: Noncooperative Stochastic Games. *Ann. Math. Statist.*, Vol. **42**, No. **6** (1971), 1930-1935.

[27] R.E. Steuer: *Multiple Criteria Optimization: Theory, Computation and Applications*. John Wiley and Sons, New York, 1986.

[28] K. Tanaka: On Some Vector Valued Markov Game. *Japan J. Appl. Math.* **2** (1985), 293-308.

[29] C.C. White, K. Kim: Solution Procedures for Vector Criterion Markov Decision Processes. *J. Large Scale Systems* **1** (1980), 129-140.

[30] P.-L. Yu: Cone Convexity, Cone Extreme Points and Nondominated Solutions in Decision Problems with Multiobjectives. *J. Optim. Theory Appl.* **14** (1974), 319-377.

[31] P.-L. Yu: *Multiple-Criteria Decision Making*. Plenum Press, New York, 1985.

Versicherung

Hiermit versichere ich, dass ich die vorliegende Arbeit ohne unzulässige Hilfe Dritter und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe; die aus fremden Quellen direkt oder indirekt übernommenen Gedanken sind als solche kenntlich gemacht. Die Arbeit wurde bisher weder im Inland noch im Ausland in gleicher oder ähnlicher Form einer anderen Prüfungsbehörde vorgelegt.

Die vorgelegte Dissertation wurde am Institut für Mathematische Stochastik der Technischen Universität Dresden unter der wissenschaftlichen Betreuung von Herrn Prof. Dr. Volker Nollau angefertigt.

Dresden, den 29. Juni 2000.

gez.

Mojca Piškurić

HSSS AdminTools (c) 2001, last visited: Tue Sep 04 11:55:56 GMT+02:00 2001